

# SKI OPTIMAL PROJECT

Tino Barras

SEPTEMBER 13, 2010

## Remerciements

Je tiens à remercier cordialement toutes les personnes ayant contribué à la réalisation de ce travail de bachelor. Une attention particulière à :

M. Michaël Schumacher, pour m'avoir encadré dans la réalisation de ce projet tout au long de ces deux mois. Merci pour les précieux conseils prodigués, pour le suivi du développement et les idées de fonctionnalités du logiciel.

M. Marc Revilloud, ancien stagiaire à l'institut d'informatique de gestion, qui a développé le logiciel Juste Neige. Merci pour sa disponibilité, bien que distante, mais très efficace et bienvenue.

M. Marut Doctor, pour m'avoir fourni tout ce dont j'avais besoin pour le développement du logiciel Ski Optimal, et pour m'avoir servi d'intermédiaire pour contacter la station de ski de Verbier.

M. Jean-Christophe Loubier, pour m'avoir aidé à définir le cahier des charges de base, et pour avoir suivi la conception du logiciel et l'évolution des fonctionnalités.

# Table des matières

|  |           |
|--|-----------|
| <b>PRÉSENTATION DU TRAVAIL .....</b>   | <b>4</b>  |
| <b>1.1 CONTEXTE DU PROJET .....</b>  | <b>4</b>  |
| <b>1.2 DESCRIPTION.....</b>  | <b>4</b>  |
| <b>1.3 NOTA BENE .....</b>   | <b>4</b>  |
| <b>2 GESTION DE PROJET.....</b>  | <b>5</b>  |
| <b>2.1 PLANNING ET SUIVI DU PROJET.....</b>                                  | <b>5</b>  |
| <b>2.2 RÉALISATION .....</b>   | <b>6</b>  |
| 2.2.1 Semaine 1 .....  | 6         |
| 2.2.2 Semaine 2 .....  | 6         |
| 2.2.3 Semaine 3 .....  | 6         |
| 2.2.4 Semaine 4 .....  | 6         |
| 2.2.5 Semaine 5 .....  | 6         |
| 2.2.6 Semaine 6 .....  | 6         |
| 2.2.7 Semaine 7 .....  | 6         |
| 2.2.8 Semaine 8 .....  | 6         |
| <b>2.3 SAUVEGARDES .....</b>   | <b>7</b>  |
| <b>3 CHOIX TECHNOLOGIQUES .....</b>  | <b>8</b>  |
| <b>3.1 SYSTÈME MULTI-AGENT.....</b>  | <b>8</b>  |
| <b>4 ANALYSE .....</b>   | <b>9</b>  |
| <b>4.1 INTRODUCTION .....</b>  | <b>9</b>  |
| <b>4.2 SIMULATION PERMANENTE .....</b>                                       | <b>9</b>  |
| <b>4.3 SIMULATION JOURNALIÈRE.....</b>                                       | <b>9</b>  |
| <b>4.4 SIMULATION DES ACCIDENTS .....</b>                                    | <b>9</b>  |
| <b>5 CHOIX DE DÉVELOPPEMENT .....</b>  | <b>10</b> |
| <b>6 DÉVELOPPEMENT .....</b>   | <b>11</b> |
| <b>6.1 ARCHITECTURE LOGICIELLE .....</b>                                     | <b>11</b> |
| <b>6.2 DONNÉES D'ENTRÉE .....</b>  | <b>12</b> |
| 6.2.1 Données à obtenir de la station de ski .....                           | 12        |
| <b>6.3 STRUCTURE D'UN SCÉNARIO .....</b>                                     | <b>14</b> |
| 6.3.1 Fichier "Config.xml" .....   | 14        |
| 6.3.2 Fichier "Station.xml" .....  | 14        |
| 6.3.3 Fichier "VolumeSkiers.dbf" .....                                       | 15        |
| <b>6.4 FONCTIONNALITÉS EXISTANTES DANS LE LOGICIEL JUSTE NEIGE .....</b>     | <b>16</b> |
| 6.4.1 Fonctionnalités conservées .....                                       | 16        |
| 6.4.2 Fonctionnalités supprimées .....                                       | 21        |
| <b>6.5 FONCTIONNALITÉS IMPLÉMENTÉES.....</b>                                 | <b>22</b> |
| 6.5.1 Introduction.....  | 22        |
| 6.5.2 Interface utilisateur .....  | 22        |
| 6.5.3 Scheduler.....   | 23        |
| 6.5.4 Informations pour une remontée mécanique .....                         | 24        |
| 6.5.5 Rectification des comptages SkiData pour une remontée principale ..... | 25        |
| 6.5.6 Agents quittant la station de ski.....                                 | 26        |
| 6.5.7 Plans d'action .....   | 28        |
| 6.5.8 Files d'attente.....   | 30        |
| 6.5.9 Encombrement de la station .....                                       | 30        |
| 6.5.10 Temps de remontée .....   | 31        |
| 6.5.11 Sauvegardes des modifications .....                                   | 32        |
| 6.5.12 Ajout d'une nouvelle remontée mécanique.....                          | 32        |

|             |  |           |
|-------------|--|-----------|
| 6.5.13      | Création d'un fichier de statistiques .....  | 33        |
| 6.5.14      | Evolution des agents .....   | 34        |
| 6.5.15      | Vérification des données SkiData .....   | 34        |
| 6.5.16      | Mouvement des remontée mécaniques .....  | 35        |
| <b>7</b>    | <b>ANALYSE DE PERFORMANCE .....</b>  | <b>35</b> |
| <b>8</b>    | <b>UTILISATION .....</b>   | <b>36</b> |
| <b>9</b>    | <b>DÉVELOPPEMENTS FUTURS.....</b>  | <b>38</b> |
| 9.1.1       | Modification du nombre total d'agents lors de l'ajout d'une nouvelle remontée..... | 38        |
| 9.1.2       | Génération d'un film de simulation.....  | 38        |
| 9.1.3       | Insertion des données de comptage horaire par l'utilisateur .....                  | 38        |
| 9.1.4       | Calcul automatique du dénivelé et de la longueur d'une installation.....           | 38        |
| 9.1.5       | Ajout de restaurants et de parkings sur le domaine skiable .....                   | 38        |
| 9.1.6       | Indications sur les pistes.....  | 38        |
| 9.1.7       | Coût d'un remblai de piste .....   | 39        |
| 9.1.8       | Oscillation, performances.....   | 39        |
| 9.1.9       | Agents quittant la station .....   | 39        |
| 9.1.10      | Etalonnage du logiciel.....  | 40        |
| 9.1.11      | Ajout de règles de vérification des données SkiData .....                          | 40        |
| 9.1.12      | Traduction du logiciel .....   | 40        |
| <b>10</b>   | <b>PROBLÈMES RENCONTRÉS .....</b>  | <b>40</b> |
| <b>10.1</b> | <b>DÉPLACEMENT DES AGENTS SUR LES PISTES .....</b>                                 | <b>40</b> |
| <b>11</b>   | <b>CONCLUSION .....</b>  | <b>41</b> |
| <b>12</b>   | <b>BIBLIOGRAPHIE ET SOURCES .....</b>  | <b>42</b> |
| <b>13</b>   | <b>TABLES DES ILLUSTRATIONS.....</b>   | <b>43</b> |
| <b>14</b>   | <b>TABLE DES TABLEAUX .....</b>  | <b>44</b> |
| <b>15</b>   | <b>TABLE DES ANNEXES.....</b>  | <b>45</b> |

# Présentation du travail

## 1.1 Contexte du projet

Lors de la création ou de la modification d'un domaine skiable, la question des flux et de leur gestion est un paramètre extrêmement important. En effet, les stations de ski existantes sont rapidement dépassées, et doivent sans cesse se mettre à jour, afin de satisfaire la clientèle toujours plus exigeante. Il s'agit donc pour les stations de ski de s'adapter continuellement aux pratiques nouvelles et à l'évolution des équipements.

Les approches actuelles de création ou de modification de station de ski reposent sur un savoir-faire indéniable, mais qui ne peut pour l'instant pas prendre en compte la totalité du système, particulièrement quand il s'agit de l'optimiser dans un espace multidimensionnel. L'outil de simulation Ski Optimal peut donc seconder efficacement les experts pour, notamment, faire évoluer un système existant vers plus d'efficacité.

## 1.2 Description

L'outil Ski Optimal sert principalement à améliorer un système existant, de façon à permettre un écoulement plus efficace des skieurs sur les pistes. Ceci aura comme conséquence de réduire le temps d'attente aux remontées mécaniques et permettra une meilleure expérience du ski.

L'outil de simulation, dans une approche d'optimisation, permet donc d'identifier les nœuds et les arcs d'un réseau de pistes de ski qui posent problème au niveau de l'écoulement de skieurs.

Différents indicateurs plus ou moins complexes peuvent être calculés pour qualifier la qualité du réseau. L'utilisateur peut modifier les propriétés des remontées mécaniques à sa guise. Procéder de cette façon est indéniablement un avantage car l'outil permet de multiples tentatives de modifications, de façon simple et interactive. L'utilisateur peut construire la solution qui lui convient le mieux. Ensuite, l'outil permet d'identifier les impacts de ces modifications sur l'ensemble de la station de ski. Et donc de traiter les effets de l'implantation d'une nouvelle remontée mécanique, en modifiant certains tracés par exemple.

Le logiciel permet également d'identifier la répercussion sur les temps d'attente aux autres différentes remontées de façon à éventuellement planifier des remplacements futurs ou encore mettre en place des stratégies de terrains destinés à pallier aux désagréments engendrés par cette nouvelle infrastructure. Ce dernier point engendre un travail considérable pour les bureaux d'études et pourrait être fortement amélioré avec cet outil. (source : Use Case Scenarii, J-C Loubier, 2010)

## 1.3 Nota bene

Le présent document a été rédigé afin de correspondre au masque de document Word utilisé pour la rédaction des projets de l'institut d'informatique de gestion.

## 2 Gestion de projet

### 2.1 Planning et suivi du projet

Au début du projet, un planning prévisionnel hebdomadaire a été établi, au format Microsoft Excel. Ce planning décrit dans les grandes lignes les tâches à effectuer pour chaque semaine de développement. Le voici ci-dessous.

|               |      |   |
|---------------|------|---|
| Analyse       | Sem1 | Définir un cahier des charges détaillé présentant plusieurs solutions de développement différentes.<br>Etudier la faisabilité des différentes solutions de développement en analysant le projet existant (analyse + tests).<br>Créer un raster <sup>1</sup> personnel, afin de tester les nouvelles fonctionnalités (terrain, pistes, remontées)  |
|               | Sem2 | En fonction de la solution retenue, définir un plan relativement détaillé des différents développement à apporter au logiciel (cahier des charges)  |
| Développement | Sem3 | Finaliser le cahier des charges, définir les développements à effectuer dans le cadre de mon travail de bachelor.<br>Début du développement, création d'une station de ski "factice".   |
|               | Sem4 | Développement du scheduler <sup>2</sup> .<br>Nombre d'agents en fonction des données de la base de données.<br>Prise en compte des jours types (beau et chaud, beau et froid, pluie, neige).<br>Rectifier le nombre de comptages, en fonction des agents entrant/sortant de la station, faire entrer les agents en fonction des données de la base de données et des remontées principales. |
|               | Sem5 | Gérer le temps de simulation, pour qu'il soit le plus réel possible (vitesse des agents).<br>Agents empruntant les remontées mécaniques (vitesse de la remontée)  |
|               | Sem6 | Files d'attente, indications à l'écran<br>Encombrement des pistes (indicateur graphique et statistique)   |
| Rapport       | Sem7 | Fichier de statistiques<br>Rapport de stage à mettre à jour<br>Relecture du code, nettoyage, commentaires, débogage.  |
|               | Sem8 | Relecture du code, nettoyage du code, commentaires, débogage.<br>Finalisation du rapport de stage, impression et reliure des documents.   |

Tableau 1 : Planning hebdomadaire prévisionnel

<sup>1</sup> Coordonnées géographiques de la surface praticable pas les skieurs.

<sup>2</sup> "Horloge" qui détermine la fréquence à laquelle est exécutée une action.

## 2.2 Réalisation

### 2.2.1 Semaine 1

Rencontre avec les différents intervenants du projet, à savoir M. Michael Schumacher<sup>3</sup>, M. Marut Doctor<sup>4</sup>, M. Marc Revilloud<sup>5</sup>, M Jean-Christophe Loubier<sup>6</sup>, afin de poser les bases du projet. Début de la rédaction du cahier des charges. Définition de plusieurs solutions de développement différentes. Lecture de la documentation concernant le logiciel Juste Neige.

### 2.2.2 Semaine 2

Analyse des différentes solutions de développement possibles. Regroupement en deux solutions distinctes. Retenue d'une solution de développement. Rédaction du cahier des charges.

### 2.2.3 Semaine 3

Choix définitif des fonctionnalités à implémenter, clôture du cahier des charges. Création d'une station "factice" pour permettre un développement davantage aisé. Développement de la modification des données d'une remontée mécanique. Enregistrement des modifications au format ".sop" (Ski Optimal Project). Développement des agents<sup>7</sup> sortant au sommet d'une remontée mécanique.

### 2.2.4 Semaine 4

Développement des agents entrant et sortant de la station, ainsi que des agents empruntant les remontées mécaniques. Développement des files d'attente au départ des remontées mécaniques. Simulation du temps (scheduler), gestion des heures et des jours types. Développement de l'algorithme rectifiant le nombre d'agents obtenu des comptages SkiData, pour une remontée mécanique principale. Mise en place du serveur SVN.

### 2.2.5 Semaine 5

Modification de la lecture du fichier DBF. Travail sur la station de Verbier. Récolte des données réelles sur les remontées mécaniques. Boutons "play", "pause", "stop". Modification de la vitesse de simulation. Correction de divers bugs liés à la station de Verbier.

### 2.2.6 Semaine 6

Développement d'une fonction qui contrôle les données de la base de données. Génération d'un fichier de statistiques durant la simulation, au format Microsoft Excel. Tests de performances. Correction de divers bugs.

### 2.2.7 Semaine 7

Coloration des pistes en fonction de la densité des agents. Commentaires dans le code, nettoyage du code. Début de la rédaction du rapport final.

### 2.2.8 Semaine 8

Rédaction du rapport final, commentaires et nettoyage du code, correction des derniers bugs. Impression et reliure des documents.

---

<sup>3</sup> Institut d'Informatique de Gestion, HES-SO Valais, Sierre

<sup>4</sup> Institut de Tourisme, HES-SO Valais, Sierre

<sup>5</sup> Institut d'Informatique de Gestion, HES-SO Valais, Sierre (Développeur du logiciel Juste Neige)

<sup>6</sup> Institut de Géomatique et d'Analyse du Risque, Université Lausanne

<sup>7</sup> Entité informatique représentant un skieur dans la réalité

## 2.3 Sauvegardes

Lors de la réalisation du projet, des sauvegardes journalières du code et de la documentation ont été effectuées grâce au logiciel DropBox<sup>8</sup>. Ce logiciel permet notamment de synchroniser des fichiers sur plusieurs ordinateurs et d'accéder aux fichiers via une interface web.

Pour compléter cette solution, des sauvegardes de code ont été effectuées, lors de chaque version fonctionnelle, soit plusieurs fois par jour. Ceci grâce à un serveur SVN<sup>9</sup> en ligne, se composant d'un espace gratuit de stockage gérant le SVN : ProjectLocker<sup>10</sup>, et d'un plugin<sup>11</sup> Eclipse : Subversive<sup>12</sup>. Ci-dessous, un aperçu d'une sauvegarde effectuée lorsque le code est fonctionnel.

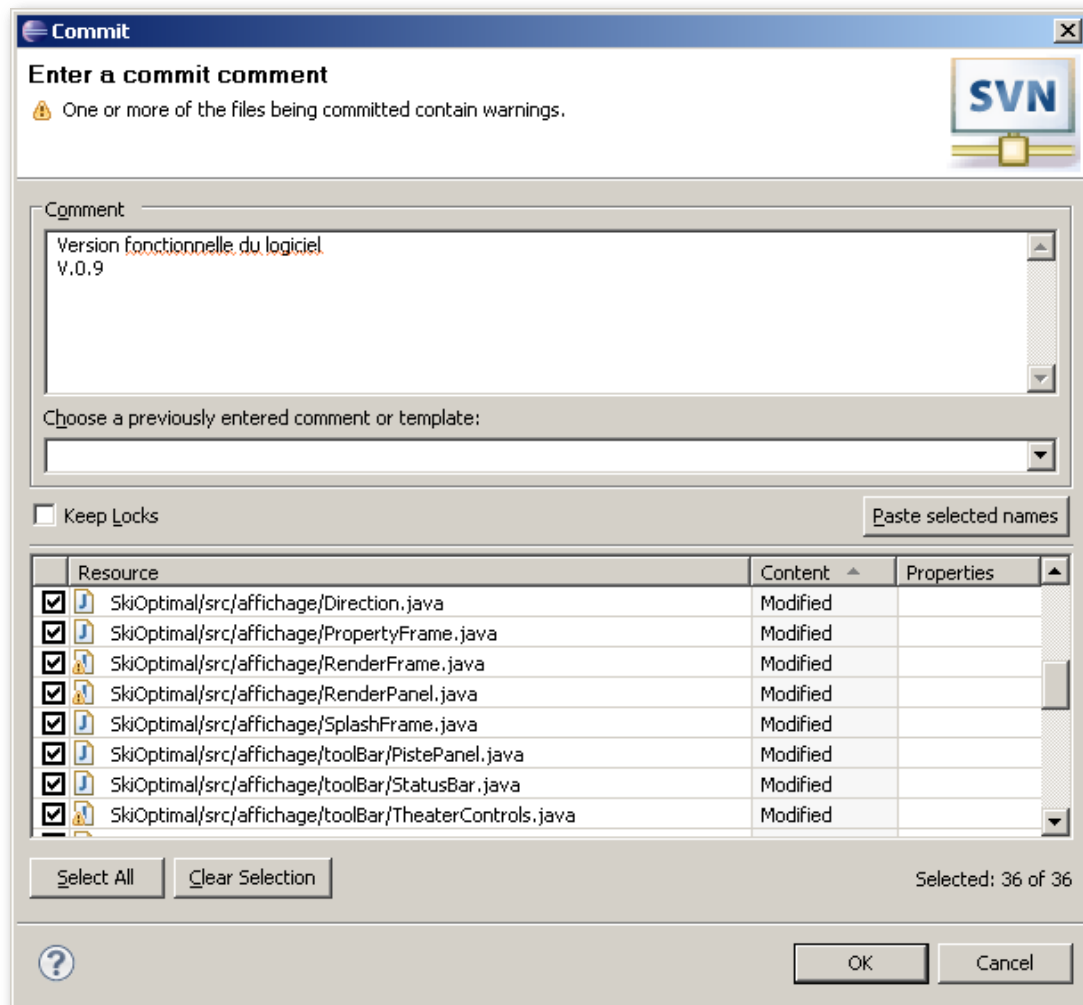


Figure 1 : Sauvegarde d'une version fonctionnelle du code.

L'utilisation de telles sauvegardes est extrêmement pratique, notamment dans le cas où l'utilisateur veut retrouver une version antérieure du code.

<sup>8</sup> Service de stockage de fichiers, <http://www.dropbox.com/>

<sup>9</sup> Système de gestion de versions

<sup>10</sup> Espace de stockage SVN, <http://projectlocker.com/>

<sup>11</sup> Module d'extension

<sup>12</sup> Plugin SVN pour Eclipse, <http://www.eclipse.org/subversive/>



### 3 Choix technologiques

Le logiciel Ski Optimal étant développé sur la base du logiciel existant Juste Neige, les choix technologiques ont été similaires à ce dernier, à savoir JAVA comme langage de programmation général. Les éléments graphiques sont codés également en JAVA, grâce à la librairie SWING. A la fin de la simulation, un fichier de statistiques au format Microsoft Excel est créé. La génération de ce document est effectuée à l'aide de la librairie JXL<sup>13</sup> intégrée au projet Ski Optimal. Les scénarios sont au format ZIP<sup>14</sup>, contenant différents fichiers au format XML<sup>15</sup> et DBF<sup>16</sup>. Ces choix ont simplement été repris du logiciel existant Juste Neige.

Le logiciel Ski Optimal intègre la notion multi-agent (explication ci-dessous) pour simuler le déplacement des agents sur le domaine skiable.

#### 3.1 Système multi-agent

Un système multi-agents peut être défini par un ensemble de processus fonctionnant simultanément, échangeant des informations entre eux et partageant des ressources communes. Ces processus sont appelés agents. Dans notre cas, il s'agit de plusieurs agents évoluant en même temps sur les pistes, mais n'interagissant pas entre eux directement. En effet, dans le logiciel Ski Optimal, un agent suit un plan d'action<sup>17</sup> qui lui est attribué au départ de son parcours, et ne modifie pas son itinéraire durant son évolution sur les pistes. Ce fonctionnement est expliqué plus bas dans le présent document.

---

<sup>13</sup> Java Excel API, <http://jexcelapi.sourceforge.net/>

<sup>14</sup> Format d'archivage et de compression, sans perte de qualité

<sup>15</sup> Langage de balisage servant à stocker et/ou transférer des données de type texte, structurées en champs arborescents (eXtensible Markup Language)

<sup>16</sup> Fichier de base de données (Data Base File)

<sup>17</sup> Suite de points décrivant l'itinéraire d'un agent

## 4 Analyse

### 4.1 Introduction

Durant la phase d'analyse du projet, plusieurs solutions se sont présentées, selon le type de simulation désiré. Il a fallu choisir entre le développement d'une solution de simulation permanente ou journalière. Ce choix déterminerait par la suite l'utilisation de données de comptages SkiData<sup>18</sup> réelles ou non.

### 4.2 Simulation permanente

La simulation permanente permet de faire évoluer un nombre d'agents fixe sur le domaine skiable, de manière continue. Elle est axée sur l'influence du flux de skieurs en fonction du débit de chaque remontée mécanique, de son emplacement et du type de pistes. Il tient compte également de l'ajout de moyens de ralentissement tels que des indications de temps d'attente, des goulots d'étranglement ou des filets de ralentissement au long des pistes.

Ce type de développement permet de suivre l'évolution des temps d'attente aux remontées principale selon différents scénarios :

- Toutes les installations ouvertes, quantité maximale d'agents sur le domaine,
- Quelques installations ouvertes, quantité moyenne d'agents sur le domaine.

Ce développement permet donc de visualiser les temps d'attente aux installations mécaniques lorsque la station fonctionne à plein régime. Il est possible également de déterminer quelles remontées mécaniques peuvent être arrêtées lorsque le nombre d'agents sur la station n'est pas au maximum.

Des réglages de débit et de vitesse peuvent être effectués par l'utilisateur sur les différentes installations mécaniques, afin de mesurer l'impact de tels changements sur les flux de skieurs et les temps d'attente aux départs des différentes installations. Des remontées mécaniques peuvent également être rajoutées, et les tracés de pistes modifiés.

### 4.3 Simulation journalière

La simulation journalière permet de faire évoluer un nombre d'agents sur le domaine, tel qu'observé dans la réalité. Ce développement, se basant sur des données de comptage SkiData, est axé sur l'influence du flux de skieurs, en fonction du débit de chaque remontée mécanique et de sa position dans l'espace.

Ce choix de développement permet de suivre l'évolution durant la journée du temps d'attente au départ de chaque installation, en fonction du nombre de skieur empruntant les remontées mécaniques. Il permet également de suivre l'évolution du flux de skieurs sur chaque tronçon de piste.

Tout comme la simulation permanente, des réglages de débit et de vitesse peuvent être effectués aux différentes installations mécaniques, afin de mesurer l'impact de ces changements sur les flux de skieurs et les temps d'attente aux départs des installations. Des remontées mécaniques peuvent également être rajoutées, et les tracés de pistes modifiés.

### 4.4 Simulation des accidents

Lors de l'étude, nous avons imaginé également la possibilité de simuler les accidents, afin de les prévenir et d'instaurer davantage de sécurité sur les pistes. En l'état de la technologie actuelle, il n'est pas possible de prévoir des accidents. Ceci demande trop de ressources logicielles. D'autant plus, nous manquons de statistiques précises et géo-localisées<sup>19</sup> concernant les accidents. Il s'avère donc impossible d'en extraire des facteurs accidentogènes.

---

<sup>18</sup> Nom du système de comptage, situé au fond des remontées mécaniques

<sup>19</sup> Localisée géographiquement

## 5 Choix de développement

Suite aux différentes discussions avec les personnes engagées dans le projet, les fonctionnalités du logiciel Ski Optimal ont été définies et arrêtées. Le logiciel se destine principalement aux personnes qui ont pour tâche de concevoir ou modifier un domaine skiable existant. Pour ce faire, une simulation journalière doit être effectuée, afin de mesurer les flux de skieurs et les temps d'attente aux différentes installations. C'est donc le développement de type "journalier" qui a été retenu. C'est sur cette base que le cahier des charges du travail a été établi. Le cahier des charges est annexé au présent document.

Le logiciel Ski Optimal doit pouvoir fournir un aperçu visuel de l'encombrement d'un domaine skiable sur une journée type, en fonction de la météorologie du jour (ensoleillé, nuageux, neigeux, pluvieux).

La simulation est une véritable aide à la décision pour la personne en charge du design de la station. Cette personne cherche par conséquent à optimiser des valeurs telles que :

- Le temps d'attente au départ des remontées mécaniques, qui influence le temps passé à skier,
- Le moment de puissance <sup>20</sup> de chaque installation,
- Choisir un compromis adéquat entre le prix d'une modification ou d'un ajout de remontée mécanique et les répercussions sur le domaine skiable, en matière de flux de skieurs.

L'utilisateur doit également être en mesure de modifier, si besoin, le domaine skiable. Cette modification peut se traduire par :

- La modification d'un tracé ou d'un type de piste,
- L'ajout, la suppression, le déplacement d'une ou de plusieurs remontées mécaniques,
- La simulation de la fermeture d'une ou de plusieurs remontée(s) mécanique(s),
- L'ajout, la modification de point d'attraction sur le domaine : restaurants, parkings...,
- L'ajout d'indications de temps d'attente à différents endroits de la piste, afin d'influencer sur le flux de skieurs.

Afin de calibrer au mieux le logiciel, les données de la station de Verbier ont été choisies. Ce choix a été fait car nous avons en notre possession pratiquement toutes les données utiles pour simuler une journée de fréquentation sur cette station de ski, étant donné que le logiciel de base, Juste Neige, a été également développé en partenariat avec cette station.

Le type de simulation choisie permet à l'utilisateur de visualiser l'évolution des flux de skieurs de deux manières :

- Agents parcourant les pistes,
- Indications de densité pour chaque tronçon de piste, dépendant du nombre d'agents et de la largeur de la piste (coloration du tronçon de piste).

Sont affichées également des données sur les files d'attente à chaque installation, qui évoluent au fil de la simulation :

- Temps d'attente en minutes,
- Nombre de personnes dans la file,
- Barre visuelle changeant de couleur en fonction du nombre d'agents dans la file.

A côté de chaque remontée mécanique sont affichés les différentes informations la concernant, ainsi que le nombre d'agents qui l'empruntent à chaque instant de la simulation.

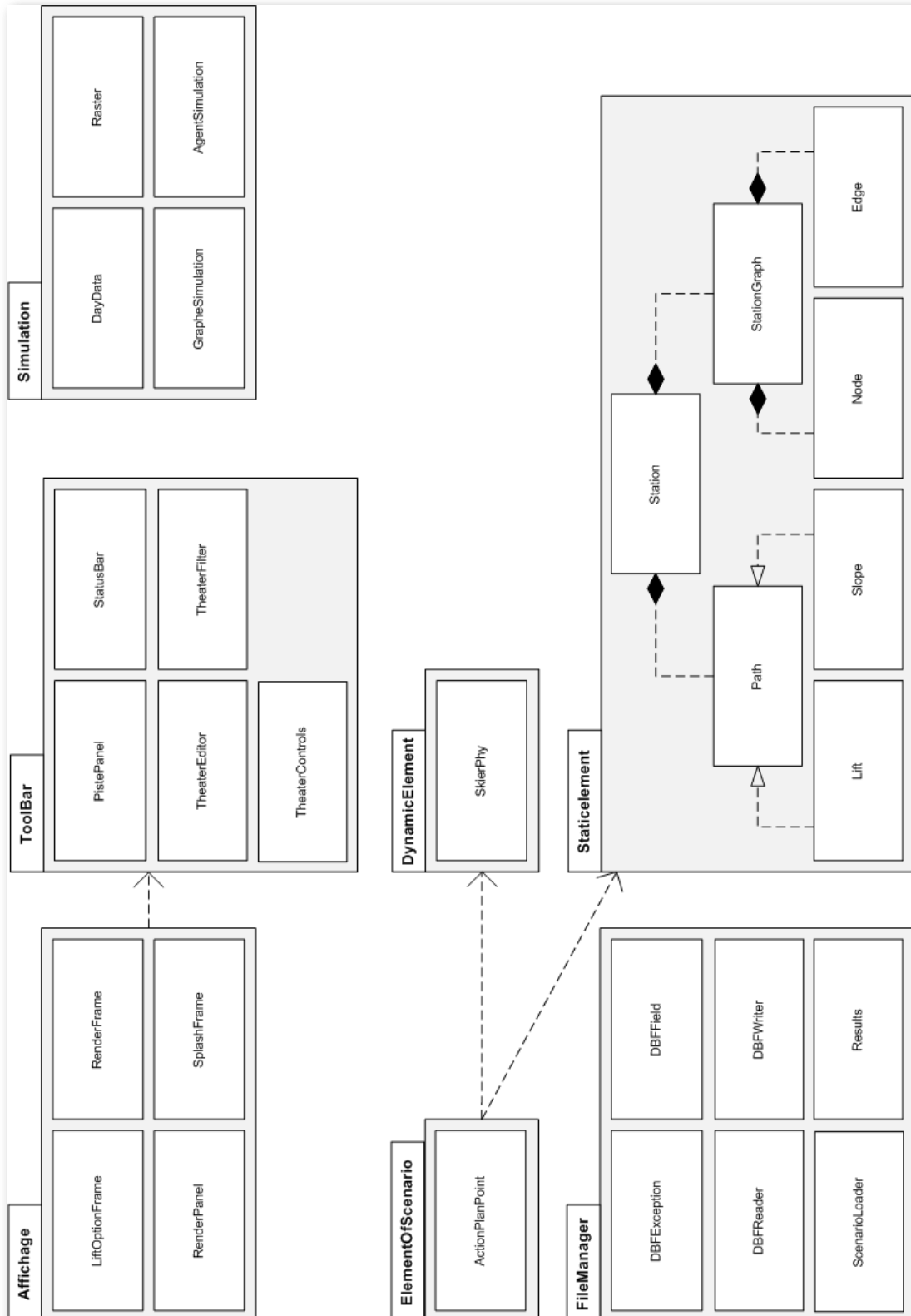
---

<sup>20</sup> Le moment de puissance est une valeur utilisée par les remontées mécaniques, afin de calculer l'efficacité des remontées mécaniques. Il se calcule ainsi : produit de la dénivellation (en kilomètres) par le débit (en personnes / heure)

## 6 Développement

### 6.1 Architecture logicielle

Le schéma suivant représente l'architecture du logiciel Ski Optimal.



Le schéma ci-dessus représente les classes les plus importantes, contenues dans les différents paquets du code JAVA de l'application Ski Optimal. Il est possible de visualiser les principales dépendances (flèches ouvertes), héritages (flèches fermées) et associations (losange noir) entre les différentes classes.

Le logiciel Ski Optimal utilise l'héritage, notamment pour la classe SkierPhy, qui hérite de l'interface Skier, et pour les classes Slope et Lift, qui étendent la classe Path. Voici un schéma :

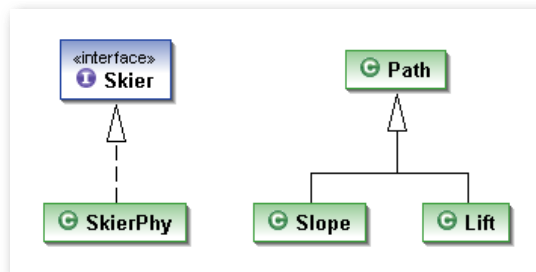


Figure 2 : Classes avec héritage

Le logiciel n'intègre pas de pattern<sup>21</sup> de programmation particulier. Le diagramme de séquence (utilisation) est illustré plus bas dans ce document (voir [Utilisation](#)).

## 6.2 Données d'entrée

Le logiciel Ski Optimal fonctionne en s'appuyant sur des données réelles. Elles sont contenues dans différents fichiers, qui sont ensuite compressés ensembles pour former un fichier de scénario.

Un fichier de scénario est donc un simple fichier compressé, appelé "NomDeLaStation.sop". L'abréviation "sop" est utilisée pour : SkiOptimalProject. Voici les différents fichiers contenus dans un fichier de scénario :

- Le raster de design de la station au format dbf (Raster1m.dbf)
- La configuration générale au format xml (Config.xml)
- La configuration de la station au format xml (Station.xml)
- Le raster de l'altitude de la station au format dbf (Alti5m.dbf)
- Le volume de skieurs, pour une journée type, par installation et par tranche horaire<sup>22</sup> (VolumeSkiers.dbf)

Ces fichiers sont expliqués en détail dans la section [Structure d'un scénario](#).

### 6.2.1 Données à obtenir de la station de ski

#### 6.2.1.1 Introduction

Pour faire fonctionner le logiciel correctement et au plus proche de la réalité, il est indispensable de recueillir certaines données concernant la station de ski simulée. Deux types de données sont requis. Des comptages de skieurs recueillis par les bornes SkiData situées au départ des remontées mécaniques, et les paramètres de chaque remontée mécanique.

---

<sup>21</sup> Modèle de conception de classes

<sup>22</sup> Par tranche horaire, nous entendons les différentes heures d'exploitation du domaine skiable durant la journée, dès l'ouverture de la station (07h00) à la fermeture des installations (16h00).

### 6.2.1.2 Données SkiData

Nous entendons par données SkiData, les comptages effectués lorsqu'un skieur passe le portique généralement installé au départ d'une installation mécanique. Il se peut qu'une installation ne dispose pas d'un tel portique. Dans ce cas, les données SkiData sont estimées au préalable.

Pour la simulation Ski Optimal, certaines données doivent être insérées dans le fichier "VolumeSkiers.dbf", contenu dans le fichier de scénario. Deux types de données sont requis :

- Nombre de remontées comptabilisées sur chaque remontée mécanique, heure par heure (07h-08h, 08h-09h, ...),
- Nombre de skieurs au total présents sur la station, pour chaque tranche horaire de la journée.

Dans le cas de l'utilisation des jours types, il faut obtenir de la part de la station, les comptages SkiData en fonction de la météo du jour en question :

- Ensoleillé et chaud (+ de 3°C à mi-altitude),
- Ensoleillé et froid (- de 3°C à mi-altitude),
- Précipitations sous forme de pluie,
- Précipitations sous forme de neige.

Ces données sont ensuite introduites dans le fichier "VolumeSkiers.dbf" (voir [Fichier "VolumeSkiers.dbf"](#)).

### 6.2.1.3 Informations sur les remontées mécaniques

Le logiciel se base sur les données réelles des remontées mécaniques. Ces différents paramètres, obtenus de la part de la station de ski sont les suivants :

- Nom de la remontée mécanique,
- Débit nominal de la remontée en mètres par seconde,
- Pourcentage d'utilisation maximale<sup>23</sup>,
- Longueur de l'installation en mètres,
- Dénivelé de la remontée mécanique en mètres,
- Vitesse de l'installation en mètres par seconde,
- Remontée principale ou non<sup>24</sup>.

Ces données sont ensuite introduites directement dans le logiciel, via l'interface à disposition. (voir [Informations pour une remontée mécanique](#))

### 6.2.1.4 Autres informations nécessaires au logiciel

Pour garantir le fonctionnement du logiciel, d'autres données sont nécessaires, comme des informations sur les types de pistes, leur tracé, leur largeur etc... Ces données sont les suivantes :

- Plan des pistes, autrement dit un raster (voir [Raster](#)),
- Tracé des pistes (voir [Plan des pistes](#)),
- Carte d'altitude de la station de ski (voir [Structure d'un scénario](#)).

Le fichier d'altitude est intégré dans le fichier de scénario.

---

<sup>23</sup> Le pourcentage maximum d'utilisation est calculé lorsqu'il y a une file d'attente au bas d'une remontée mécanique, et que la remontée mécanique tourne à plein débit. Si cette donnée n'existe pas, elle peut être estimée par l'utilisateur.

<sup>24</sup> Nous entendons par remontée principale, un point d'entrée/sortie de la station de ski.

## 6.3 Structure d'un scénario

Pour simplifier l'utilisation et pour éviter toute erreur de manipulation, il a été décidé d'unifier toutes les données propres à une station dans un fichier unique, formaté comme suit : "NomDeLaStation.sop". Ce dernier est ce que l'on appelle le *scénario*. Ce n'est en fait qu'un fichier archivé dans le format ZIP. Il comprend des fichiers au format DBF et XML:

- Alti5m.dbf : MNT<sup>25</sup> à 5m (fichier de type DBF avec les coordonnées X, Y et Z pour la hauteur),
- Config.xml : paramètres utilisateurs pour ce scénario,
- Raster1m.dbf : raster à 1m (fichier de type DBF les coordonnées X, Y des cellules du raster),
- Station.xml : graphe des pistes (fichier XML, contient l'ensemble des pistes et des remontées, ainsi que leurs points),
- VolumesSkiers.dbf : fréquentation des skieurs par remontée mécanique par heure.

Les fichiers "Config.xml", "Station.xml" et "VolumeSkiers.dbf" sont détaillés ci-dessous.

### 6.3.1 Fichier "Config.xml"

Le fichier "Config.xml" a été légèrement modifié par rapport au fichier qu'utilise le logiciel Juste Neige, afin d'intégrer les nouveaux filtres (voir [Filtre du théâtre](#)) du scénario qu'utilise le logiciel Ski Optimal :

- Densité sur les pistes,
- Files d'attente au départ des installations.

Ces filtres permettent d'afficher ou non ces informations (voir [Filtre du théâtre](#)).

### 6.3.2 Fichier "Station.xml"

Ce fichier a également été étoffé afin d'intégrer les différentes informations sur les remontées mécaniques. Il contient les éléments suivants :

- Une ou plusieurs pistes
  - Nom de la piste,
  - Tracé de la piste (coordonnées),
  - Type de la piste (couleur),
  - Sens de la piste,
  - Piste raccordée à une autre piste ou à une remontée mécanique (noeud),
  - Position en X et Y du nom.
- Une ou plusieurs remontées mécaniques
  - Nom de la remontée,
  - Type de la remontée,
  - Tracé de la remontée (coordonnées),
  - Débit nominal horaire de la remontée,
  - Définir au moins une remontée principale,
  - Vitesse de la remontée mécanique,
  - Longueur de la remontée,
  - Dénivelé de la remontée,
  - Position en X et Y des données.
- Prix d'une nouvelle remontée / modification d'une remontée (optionnel).

---

<sup>25</sup> Fichier représentant la topographie d'une zone (Modèle Numérique de Terrain)

### 6.3.3 Fichier "VolumeSkiers.dbf"

#### 6.3.3.1 Introduction

Ci-dessous sont décrites les différentes données contenues dans ce fichier, ainsi que sa structure. Les informations y sont insérées par tranche horaire. Ce choix a été fait car les données récoltées auprès de SkiData sont des comptages horaires. Une journée est par défaut divisible en dix tranches horaires, partant de 07h00 et allant jusqu'à 17h.

Le nombre d'heures simulées peut être modifié dans le code, à l'aide de la ligne suivante, contenue dans le fichier "simulation.java".

```
public static int nbHoursToSimulate = 10;
```

Nous supposons que toutes les installations ferment à 16h au plus tard.

#### 6.3.3.2 Structure du fichier

Le fichier "VolumeSkiers.dbf" contient comme clé un champ composé du jour type (voir [Jours types](#)) et de la tranche horaire correspondante. Ensuite sont listées les valeurs correspondant aux comptages sur les différentes installations mécaniques, comme illustré dans le tableau ci-dessous.

| HOURL | ATTELAS1 | ATTELAS2 | CHAUX2 | ... | TOTAL |
|-------|----------|----------|--------|-----|-------|
| 01-01 | 159      | 0        | 0      | ... | 200   |
| 01-02 | 1775     | 616      | 142    | ... | 2000  |
| 01-03 | 1971     | 1617     | 1099   | ... | 3400  |
| ...   | ...      | ...      | ...    | ... | ...   |

Tableau 2 : Structure du fichier "VolumeSkiers.dbf"

Composition du champ *HOURL* : j-n, j étant l'identifiant du jour type (01, 02, 03 ou 04), n étant l'identifiant de la tranche horaire. Voici les différents jours types possibles :

- Jour de type 1 : ensoleillé et plus de 3°C à mi-hauteur du domaine skiable
- Jour de type 2 : ensoleillé et moins de 3°C à mi-hauteur du domaine skiable
- Jour de type 3 : pluvieux et plus de 3°C à mi-hauteur du domaine skiable
- Jour de type 4 : neigeux et plus de 3°C à mi-hauteur du domaine skiable

Les données de fréquentation par remontée mécanique sont utilisées uniquement pour diriger les agents vers leur destination (voir [Initialisation des plans d'action](#)).

Dans le cas où la remontée mécanique est une remontée principale, ces données sont retravaillées avant d'être utilisées pour la détermination d'une destination (voir [Rectification des comptages SkiData pour une remontée principale](#)).



## 6.4 Fonctionnalités existantes dans le logiciel Juste Neige

### 6.4.1 Fonctionnalités conservées

#### 6.4.1.1 Introduction

Le développement du projet Ski Optimal se fait sur la base du logiciel Juste Neige. De nombreuses fonctionnalités développées dans le cadre de ce logiciel sont utilisables. Notamment les données que nous avons en notre possession concernant les pistes de ski et les remontées mécaniques de la station de Verbier.

Certaines fonctionnalités du logiciel Juste Neige sont donc conservées pour le développement de Ski Optimal. Certaines fonctionnalités sont modifiées, et évidemment, certaines sont ajoutées. Les éléments conservés et parfois modifiés sont listés et décrits ci-dessous.

#### 6.4.1.2 Raster

Le logiciel Juste Neige permet de créer et modifier un raster existant. Un raster se définit ici par les coordonnées de la surface praticable par les skieurs. Ce dernier est obtenu de la part de la station de ski, Verbier dans notre cas. Cette fonctionnalité est conservée tel quelle dans le cadre du logiciel Ski Optimal.

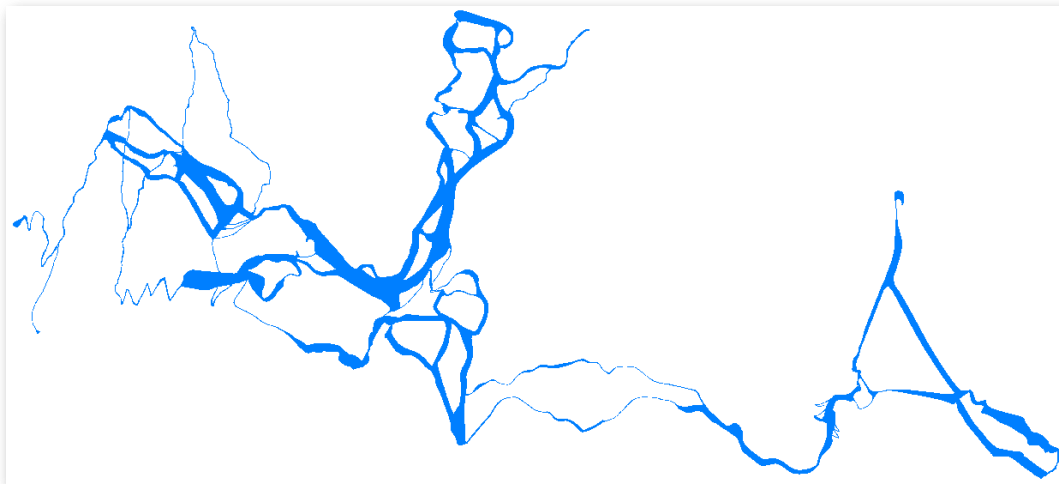


Figure 3 : Raster de la station de Verbier

#### 6.4.1.3 Plan des pistes

Dans le cas de la station de Verbier, le plan des pistes (c.f. figure 4) a été fourni par Geosat<sup>26</sup> lors de la création du logiciel Juste Neige. Ce plan est dans un format SIG<sup>27</sup>. Il a été restructuré afin d'être intégré dans l'outil.

Le logiciel Ski Optimal se base également sur ces données. Chaque piste peut ensuite être modifiée par l'utilisateur, directement dans le logiciel (voir [Création et modification de piste de ski et de remontées mécaniques](#)).

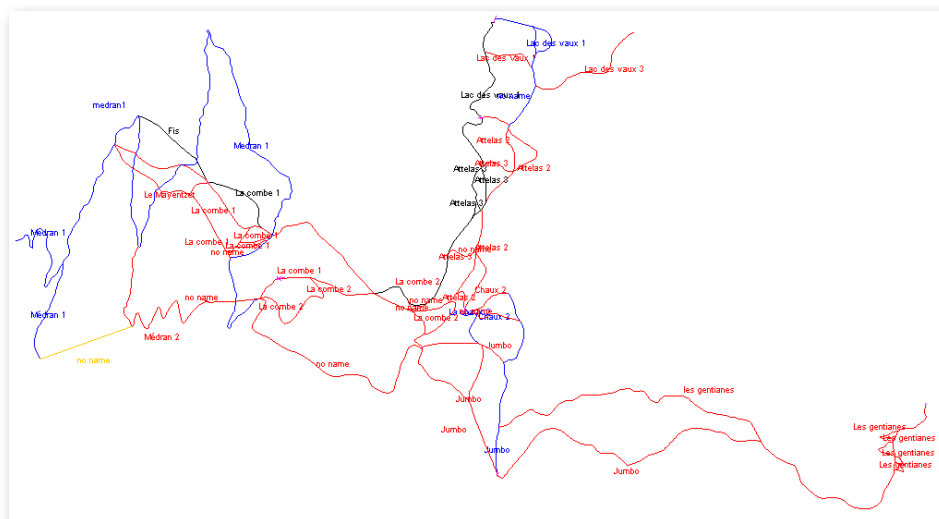


Figure 4 : Représentation du plan des pistes de Verbier

#### 6.4.1.4 Création et modification de piste de ski et de remontées mécaniques

Le logiciel Juste Neige permet de créer de nouvelles remontées mécaniques et de nouvelles pistes de ski. Le logiciel Ski Optimal conserve ces fonctionnalités (voir [Edition du théâtre](#)).

La création ou modification d'une remontée mécanique peut se traduire par la personnalisation du tracé, du nom, du débit nominal, du coefficient maximum d'utilisation, de la vitesse, de la longueur, du dénivelé, des coûts de la modification, et de la définition de cette remontée mécanique comme principale ou non (voir [Informations pour une remontée mécanique](#)).

#### 6.4.1.5 Calcul du coefficient d'attraction

Le logiciel Juste Neige est pourvu d'un algorithme permettant de calculer le coefficient d'attraction de chaque point de départ depuis le sommet d'une remontée mécanique, en fonction du nombre de skieurs ayant emprunté cette remontée mécanique dans la réalité<sup>28</sup>. Le coefficient est donc la donnée qui permet de déterminer si un départ de remontée mécanique est davantage "visé" qu'un autre, au cours d'une tranche horaire.

Cet algorithme de parcours calcule également le chemin que parcourt un agent, en fonction du type de pistes qui se présentent à lui. Le choix d'un type de piste plutôt qu'un autre se fait en fonction du jour type choisi, et bien évidemment en fonction de la destination de l'agent. Cet algorithme est repris tel quel dans le cadre du développement du logiciel Ski Optimal.

<sup>26</sup> Entreprise d'ingénieurs et de géomètres basée à Sion

<sup>27</sup> Représentation informatique d'une information géographique

<sup>28</sup> En référence aux comptages SkiData

#### 6.4.1.6 Edition du théâtre

La palette d'édition du théâtre<sup>29</sup> développée dans le logiciel Juste Neige est reprise dans le logiciel Ski Optimal. Elle est disponible en cliquant sur le lien "Theater Editor" se trouvant dans la barre de menus "View", ou en utilisant la touche du clavier "E". La création d'une piste de type verte a été supprimée (voir [Piste « verte »](#)). La palette d'édition permet d'éditer la station dans son ensemble. Deux outils de sélection sont proposés :

- Sélection des pistes : permet de sélectionner une piste du théâtre en cliquant dessus,
- Sélection raster : permet de sélectionner des cellules du raster, pour ensuite les modifier.

L'outil d'édition permet également de choisir le type de piste ou de remontée mécanique à créer. Voici un aperçu de la palette d'édition:



Figure 5 : Palette d'édition du théâtre

Une fois qu'un changement est effectué sur la station, les modifications peuvent être sauvegardées (voir [Sauvegardes des modifications](#)).

---

<sup>29</sup> Théâtre : fenêtre de simulation contenant le raster, les pistes, les remontées mécaniques, etc...

#### 6.4.1.7 Filtre du théâtre

Le filtre du théâtre développé pour le logiciel Juste Neige est également conservé pour le développement de Ski Optimal.

Ce filtre permet de sélectionner l'affichage des informations que l'utilisateur désire à l'écran. Il est disponible en cliquant sur le lien "Theater Filter" se trouvant dans la barre de menus "View", ou en utilisant la touche du clavier "F".

Le filtre se présente sous la forme d'une arborescence où chaque nœud de l'arbre correspond à un type de filtre, et où les feuilles sont les filtres. Le changement d'état d'un nœud se répercute sur les feuilles. Voici un aperçu du filtre du théâtre :

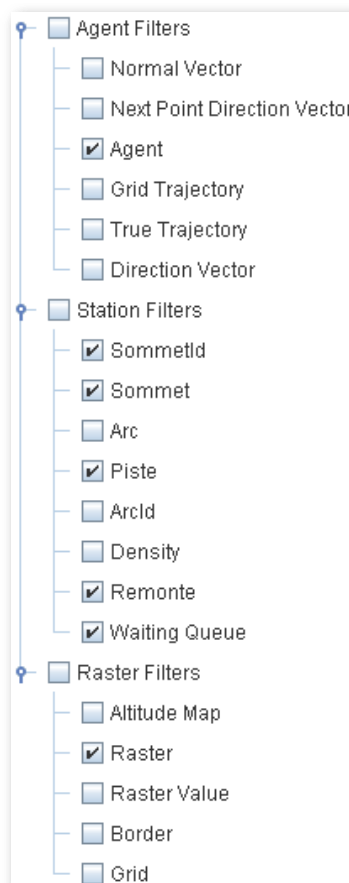


Figure 6 : Filtre du théâtre

Deux options de filtrage sont ajoutées dans le cadre du logiciel Ski Optimal. Ces deux filtres supplémentaires sont :

- Les files d'attente (voir [Files d'attente](#)),
- La densité des agents (voir [Couleur des tronçons de pistes](#)).

Une fois qu'un filtre est modifié, ces changements peuvent être sauvegardés (voir [Sauvegardes des modifications](#)), afin que lors de la prochaine ouverture du scénario, l'utilisateur puisse bénéficier de ces paramètres d'affichage.

Le filtre peut être utilisé au cours de la simulation, si elle est mise en pause ou si elle n'est pas lancée.

#### 6.4.1.8 Jours types

Le logiciel de base, Juste Neige, intègre des probabilités de choix de piste, selon la difficulté de la piste, par beau ou mauvais temps et par neige lourde ou de bonne qualité. Une étude a été réalisée pour sa conception; cela a permis de ressortir des données sous formes de règles, influençant pour un agent, le choix d'une piste plutôt qu'une autre.

Ces informations sont reprises dans le logiciel Ski Optimal. Voici les règles qui en découlent, et qui sont intégrées au logiciel, dans la classe "SimuGraph.java" :

```
if(jourType == 1) // Sunny and warm
{
    blueProba = 0.27f;
    redProba = 0.42f;
    blackProba = 0.23f;
    yellowProba = 0.08f;
} else if(jourType == 2) // Sunny and cold
{
    blueProba = 0.09f;
    redProba = 0.36f;
    blackProba = 0.25f;
    yellowProba = 0.25f;
} else if(jourType == 3) // Rain
{
    blueProba = 0.27f;
    redProba = 0.44f;
    blackProba = 0.21f;
    yellowProba = 0.08f;
} else if(jourType == 4) // Snow
{
    blueProba = 0.23f;
    redProba = 0.46f;
    blackProba = 0.21f;
    yellowProba = 0.10f;
}
```

Explications des différentes variables :

- Variable *jourType* : au lancement de chaque simulation, l'utilisateur choisit parmi les jours types proposés, celui qu'il désire (voir [Barre de simulation](#)).
- Variables *slopeProba* : probabilités de choix de tel ou tel type de piste de ski.

A ce stade du développement, nous sommes uniquement en possession des comptages pour le jour de fréquentation "record" de la station de Verbier, autrement dit seulement pour un seul jour. Le logiciel ne peut donc pas se baser sur les quatre jours types pour récupérer les comptages SkiData.

Dans le cas où les données de fréquentation pour chaque jour type sont accessibles et présents dans la base de données (voir [Fichier "VolumeSkiers.dbf"](#)), l'utilisateur veillera à modifier la ligne suivante, présente au sommet de la classe "Simulation.java"

```
public static boolean skiDataCountsForAllDayTypeExistsInDBF = false;
```

en

```
public static boolean skiDataCountsForAllDayTypeExistsInDBF = true;
```

#### 6.4.1.9 Liste des pistes

Cette fonctionnalité est purement pratique, elle permet de retrouver simplement une piste dans un domaine skiable (cf. figure 7). Elle est disponible en cliquant sur le lien "Slope Panel" se trouvant dans la barre de menus "View", ou en utilisant la touche du clavier "S". Cette liste se présente sous la forme d'une barre latérale composée de deux éléments :

- La zone de recherche au sommet de la barre qui permet de filtrer les éléments de la liste en fonction du texte introduit,
- La liste des pistes de station. Une piste peut être sélectionnée pour apparaître au premier plan dans le théâtre.

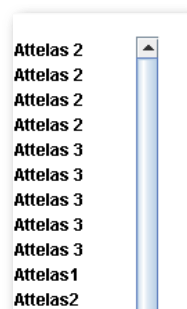


Figure 7 : Sélection des pistes

### 6.4.2 Fonctionnalités supprimées

#### 6.4.2.1 Hauteur de neige et concavité de la piste

La prise en compte de ces deux points n'influence pas le comportement des agents dans le cadre de la simulation de flux. Ils ne sont pas conservés dans le développement du logiciel Ski Optimal.

#### 6.4.2.2 Menu « Simulation »

Le menu « Simulation », qui comporte les sous-menus suivants :

- Simulation GRNN
- Simulation Météo
- Simulation Graphe
- Simulation Skieur
- Simulation Juste Neige

N'est également pas conservé dans le cadre du développement de Ski Optimal.

#### 6.4.2.3 Piste « verte »

Création d'une piste de type « Verte », n'est pas conservée pour le développement du logiciel Ski Optimal. Ce type de piste n'existe pas en Suisse. Nous ne disposons donc pas de statistiques quant au choix d'un tel type de piste, en fonction de la météorologie.

La gestion de ce type de piste reste néanmoins possible. Elle est simplement commentée dans le code de l'application.

## 6.5 Fonctionnalités implémentées

### 6.5.1 Introduction

Le travail de bachelor durant lequel a été développé le logiciel Ski Optimal est un travail exploratoire. Il permet de démontrer la faisabilité d'un tel outil de simulation pour un développement futur. C'est pourquoi certaines fonctionnalités subsistent dans la rubrique "développements futurs" (voir [Développements futurs](#)).

Ci-dessous sont listées et détaillées les fonctionnalités implémentées dans le cadre dudit travail de bachelor.

### 6.5.2 Interface utilisateur

#### 6.5.2.1 Barre de simulation

L'utilisateur dispose d'une "barre de simulation" (c.f. figure 8) qui lui permet de contrôler le déroulement de la simulation.



Figure 8 : Barre de simulation

Elle se divise en plusieurs modules distincts :

- Choix du jour type (c.f. figure 9),
- Contrôle de la simulation et choix de la vitesse de simulation (c.f. figure 10),
- Nombre de skieurs, coûts des modifications et choix du comportement des skieurs (c.f. figure 11).



Figure 9 : Choix du jour type

Les boutons du module ci-dessus permettent de choisir le jour type à simuler. Ceci va influencer la répartition des agents sur le domaine skiable, selon le type de pistes qu'ils choisissent d'emprunter. Si les données SkiData comprennent les comptages pour les quatre jours types, ce sont ces informations qui seront alors utilisées pour la simulation (voir [Jours types](#)).



Figure 10 : Contrôle de la simulation et choix de la vitesse

Le module de contrôle de simulation permet de démarrer la simulation, de la mettre en pause et de la stopper au moyen des trois boutons ci-dessus. L'utilisateur peut également choisir la vitesse de simulation. Par défaut, la vitesse est réglée sur 100x la réalité.

Le module ci-dessous affiche le nombre de skieurs présents sur la station durant chaque tranche horaire. Ce module affiche également le coût des modifications apportées.

|                  |  |
|------------------|--|
| Skiers : 0       | <input checked="" type="radio"/> straight comportement |
| Costs : 0.00 CHF | <input type="radio"/> oscillation comportement (bêta)  |

Figure 11 : Nombre de skieurs, coûts des modifications et choix du comportement des skieurs

L'utilisateur également a la possibilité de choisir le comportement des skieurs. Par défaut, les skieurs suivent plus ou moins le tracé de la piste (straight comportement). Ce comportement simule une vitesse de ski s'approchant de la réalité, mais ne permet pas visuellement d'observer l'oscillation d'un skieur. Cette option est donc par défaut utilisée afin d'augmenter les performances de la simulation.

Inversement, en choisissant le comportement oscillant des agents, l'utilisateur peut apercevoir les skieurs oscillant sur les pistes de ski. Cette option n'est pour l'instant pas optimale car la vitesse de ski ne s'approche pas de la réalité, et les performances du logiciel s'en trouvent fortement réduites (saccades). C'est donc pour l'instant une option purement visuelle (voir [Oscillation, performances](#)).

### 6.5.3 Scheduler

Le logiciel Ski Optimal permet la simulation de l'évolution des skieurs de manière journalière. Pour ce faire, la journée est découpée en plusieurs tranches horaires. Afin de simuler le temps, un scheduler est intégré dans le logiciel. L'utilisateur a la possibilité, au lancement de la simulation, de choisir la vitesse de simulation. C'est ce choix qui détermine le nombre de pas que le scheduler effectue durant les dix heures simulées. La vitesse est par défaut initialisée à 100x la réalité : 3600 pas de 10 secondes, chaque 0.1 seconde.

Le scheduler a pour tâche :

- La prise en compte des différentes tranches horaires,
- La gestion des agents qui empruntent une remontée mécanique, en fonction du débit horaire effectif de celle-ci,
- De simuler au plus proche de la réalité le temps que met un agent pour parcourir une piste,
- De simuler le temps que dure la remontée,
- D'indiquer le temps d'attente au départ de chaque installation,
- De gérer l'ensemble de la simulation au moyen de 3 boutons.



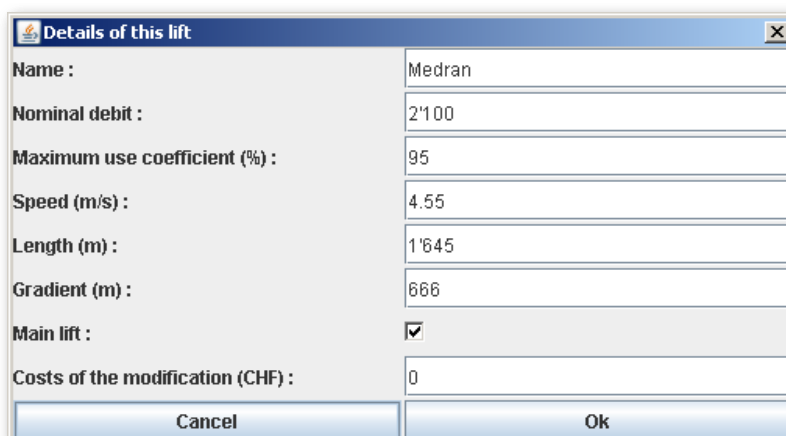
Lorsque l'utilisateur interrompt la simulation au moyen du bouton pause, la simulation se "gèle". Il peut ainsi consulter les données affichées à l'écran. A ce moment, ainsi que durant le fonctionnement de la simulation, le filtre de l'affichage peut être modifié, afin d'afficher les éléments choisis par l'utilisateur dans le théâtre de simulation.



#### 6.5.4 Informations pour une remontée mécanique

Le logiciel Ski Optimal offre la possibilité à l'utilisateur de modifier les données concernant une remontée mécanique (c.f. figure 12) dans le cas où elle existe déjà sur le domaine. L'utilisateur utilise également cette fenêtre lorsqu'il ajoute une nouvelle remontée sur la station. Ces données sont les suivantes :

- Nom de la remontée mécanique,
- Débit nominal horaire de la remontée en personnes par heure,
- Pourcentage d'utilisation maximum,
- Vitesse de la remontée mécanique en mètres secondes,
- Longueur de la remontée en mètres,
- Dénivelé de la remontée mécanique en mètres,
- Remontée principale ou non,
- Prix d'une nouvelle remontée ou d'une modification, en francs Suisses.



| Details of this lift              |                                     |
|-----------------------------------|-------------------------------------|
| Name :                            | Medran                              |
| Nominal debit :                   | 2'100                               |
| Maximum use coefficient (%) :     | 95                                  |
| Speed (m/s) :                     | 4.55                                |
| Length (m) :                      | 1'645                               |
| Gradient (m) :                    | 666                                 |
| Main lift :                       | <input checked="" type="checkbox"/> |
| Costs of the modification (CHF) : | 0                                   |
| <div>Cancel Ok</div>              |                                     |

Figure 12 : Détails d'une remontée mécanique

Pour que la simulation fonctionne convenablement, une nouvelle remontée mécanique doit être impérativement raccordée aux pistes existantes (nœuds), et respecter un sens logique (du - vers le +). Ceci peut se vérifier lorsque la remontée mécanique est sélectionnée. Un petit signe (+) se trouve à l'extrémité du tracé de la remontée. Un signe (-) se trouve à l'autre extrémité.

### 6.5.5 Rectification des comptages SkiData pour une remontée principale

Le logiciel Ski Optimal propose une simulation journalière. Il faut donc simuler les skieurs qui entrent et qui sortent de la station de ski<sup>30</sup>. Le logiciel doit adapter le nombre de skieurs obtenus des comptages SkiData en fonction du nombre de skieurs totaux, afin de rectifier les comptages SkiData.

Pour cela, nous partons du principe qu'une remontée mécanique principale<sup>31</sup> qui a comptabilisé plus de remontées qu'une autre remontée mécanique principale, est empruntée par davantage d'agents entrant dans la station. Nous supposons également que davantage d'agents quittant la station se rendent au bas de cette remontée mécanique ayant comptabilisé davantage de remontées qu'une autre. Le calcul se fait alors de manière différente, sachant que les agents entrent ou sortent de la station de ski.

Il est possible de calculer le nombre d'agents entrant et sortant de la station durant une tranche horaire en faisant la différence avec la tranche horaire précédente. C'est ce résultat qui est utilisé pour corriger le nombre de comptages SkiData d'une remontée mécanique principale, afin d'avoir un résultat cohérent.

Dans le cas où nous avons plusieurs remontées principales dans la station, le logiciel répartit les nouveaux skieurs et ceux sortant du domaine de manière proportionnelle sur ces installations mécaniques. C'est-à-dire en fonction de leurs comptages SkiData bruts. Ces calculs sont expliqués ci-dessous.

#### 6.5.5.1 Rectification des comptages lorsque des agents entrent dans la station de ski

Lorsque des skieurs entrent dans la station, leur passage est également comptabilisé par les bornes SkiData. Il faut alors soustraire ces comptages aux comptages SkiData, afin d'utiliser pour le calcul d'itinéraire seulement les agents déjà présents sur la station et empruntant cette remontée. Ceci a pour effet de corriger l'attraction de ce départ de remontée mécanique, et ainsi de mieux refléter la réalité.

Prenons le cas où nous avons deux remontées principales dans une station de ski. Nous savons que pour la période 09h-10h, il y a eu 200 nouveaux skieurs sur le domaine :

- |              |                  |                                   |
|--------------|------------------|-----------------------------------|
| • Remontée_1 | débit : 3000 p/h | comptages 09h-10h : 1500 passages |
| • Remontée_2 | débit : 4000 p/h | comptages 09h-10h : 3000 passages |

Il s'agit maintenant de calculer les coefficients par rapport aux comptages des autres remontées principales :

- |              |   |
|--------------|---|
| • Remontée_1 | coefficient d'utilisation : $1500 / (1500+3000) = 0.33$ |
| • Remontée_2 | coefficient d'utilisation : $3000 / (1500+3000) = 0.66$ |

Ensuite, cela nous donne le nombre de skieurs entrant dans la station, à chaque installation principale :

- |              |  |
|--------------|--|
| • Remontée_1 | nombre d'agents entrants : $200 * 0.33 = 67$ agents  |
| • Remontée_2 | nombre d'agents entrants : $200 * 0.66 = 133$ agents |

Nous avons maintenant attribué à chaque remontée principale, le nombre d'agents entrant dans la station de ski. Reste à modifier les comptages SkiData, afin qu'ils prennent en compte cette donnée :

- |              |   |
|--------------|---|
| • Remontée_1 | nombre d'agents pour le calcul : $1500 - 67 = 1433$ agents  |
| • Remontée_2 | nombre d'agents pour le calcul : $3000 - 133 = 2867$ agents |

Nous utilisons donc ces nouvelles données pour le calcul d'itinéraire. Le logiciel crée les nouveaux agents aux différents départs d'installation, à intervalles réguliers durant la tranche horaire. Ces intervalles sont calculés en fonction du nombre de pas exécutés durant chaque tranche horaire.

---

<sup>30</sup> Pour cette phase du projet, nous supposons que les personnes quittant la station le fassent au départ d'une remontée mécanique principale.

<sup>31</sup> Remontée mécanique par laquelle les skieurs entrent et sortent de la station

#### 6.5.5.2 Rectification des comptages lorsque des agents entrent quittent la station de ski

A l'inverse, lorsque le nombre de skieurs diminue, en fin de journée par exemple, les comptages doivent également être adaptés. Etant donné que les skieurs empruntant la remontée mécanique sont cette fois moins nombreux que les skieurs se rendant au fond de cette remontée (parking, arrêts de bus, ...), il faut ajouter ces skieurs sortant aux comptages SkiData. Ceci aura pour effet de rendre ce point plus attractif pour les agents lors de la simulation, et ainsi d'être plus proche de la réalité.

Prenons le cas où nous avons deux remontées principales dans la station de ski. Nous savons que pour la période 15h-16h, il y a eu 600 skieurs qui ont quitté le domaine skiable :

- Remontée\_1                      débit : 3000 p/h                      comptages 15h-16h: 500 passages
- Remontée\_2                      débit : 4000 p/h                      comptages 15h-16h: 700 passages

Il s'agit maintenant de calculer les coefficients par rapport aux comptages des autres remontées principales :

- Remontée\_1                      pourcentage d'utilisation :  $500 / (500+700) = 0.42\%$
- Remontée\_2                      pourcentage d'utilisation :  $700 / (500+700) = 0.58\%$

Ensuite, cela nous donne le nombre de skieurs sortant à chaque installation principale :

- Remontée\_1                      nombre d'agents sortants :  $600 * 0.42 = 252$  agents
- Remontée\_2                      nombre d'agents sortants :  $600 * 0.58 = 348$  agents

Nous avons maintenant attribué à chaque remontée principale, le nombre d'agents entrant dans la station de ski. Reste à modifier les comptages SkiData, afin qu'ils prennent en compte cette donnée :

- Remontée\_1                      nombre d'agents pour le calcul :  $500 + 252 = 752$  agents
- Remontée\_2                      nombre d'agents pour le calcul :  $700 + 348 = 1048$  agents

Nous utilisons donc ces nouvelles données pour le calcul d'itinéraire. Le nombre d'agents à supprimer est attribué à remontée mécanique. Le logiciel supprime ensuite des agents aux différents départs de remontée mécanique principale, à intervalles réguliers durant la tranche horaire (voir [Agents quittant la station de ski](#)).

#### 6.5.6 Agents quittant la station de ski

Une fois qu'une remontée mécanique principale a des agents à supprimer durant la tranche horaire, le logiciel calcule le nombre d'agents à supprimer à chaque pas de simulation. Étant donné qu'un nombre d'agents ne peut être qu'entier, la valeur suivant la virgule est conservée dans une variable appelée *agentsFloat*, afin de conserver l'information. Le calcul se fait selon le pseudo-code suivant :

```
POUR CHAQUE Remontée mécanique
  SI la remontée a des agents à supprimer ALORS
    SI la remontée n'a pas d'agentsFloat à supprimer ALORS
      agentsFloat = nb. d'agents à supprimer / nb. pas pour 1h simulée
    FIN SI
    On stocke le chiffre après la virgule d'agentsFloat
    On en retire un entier si  $\geq 1.00$ 
    On ajoute cet entier au nb. d'agents à supprimer durant ce pas
  FIN SI
FIN POUR CHAQUE
```

Illustrons le code ci-dessus avec un exemple. Prenons le cas où 27 skieurs quittent la station durant l'heure simulée, à une remontée mécanique principale. Admettons que l'heure simulée soit composée de 10 pas. Ceci nous fait 2.7 personnes par pas (27 agents / 10 pas).

- |            |                           |               |                 |               |                    |
|------------|---------------------------|---------------|-----------------|---------------|--------------------|
| • Pas 1 :  | reste : 0.7               |               |                 |               | 2 agents           |
| • Pas 2 :  | reste : $0.7 + 0.7 = 1.4$ | $\Rightarrow$ | $1.4 - 1 = 0.4$ | $\Rightarrow$ | $2 + 1 = 3$ agents |
| • Pas 3 :  | reste : $0.4 + 0.7 = 1.1$ | $\Rightarrow$ | $1.1 - 1 = 0.1$ | $\Rightarrow$ | $2 + 1 = 3$ agents |
| • Pas 4 :  | reste : $0.1 + 0.7 = 0.8$ |               |                 | $\Rightarrow$ | 2 agents           |
| • Pas 5 :  | reste : $0.8 + 0.7 = 1.5$ | $\Rightarrow$ | $1.5 - 1 = 0.5$ | $\Rightarrow$ | $2 + 1 = 3$ agents |
| • Pas 6 :  | reste : $0.5 + 0.7 = 1.2$ | $\Rightarrow$ | $1.2 - 1 = 0.2$ | $\Rightarrow$ | $2 + 1 = 3$ agents |
| • Pas 7 :  | reste : $0.2 + 0.7 = 0.9$ |               |                 | $\Rightarrow$ | 2 agents           |
| • Pas 8 :  | reste : $0.9 + 0.7 = 1.6$ | $\Rightarrow$ | $1.6 - 1 = 0.6$ | $\Rightarrow$ | $2 + 1 = 3$ agents |
| • Pas 9 :  | reste : $0.6 + 0.7 = 1.3$ | $\Rightarrow$ | $1.3 - 1 = 0.3$ | $\Rightarrow$ | $2 + 1 = 3$ agents |
| • Pas 10 : | reste : $0.3 + 0.7 = 1$   | $\Rightarrow$ | $1 - 1 = 0$     | $\Rightarrow$ | $2 + 1 = 3$ agents |

Au total, nous avons bel et bien 27 agents qui ont quitté la station de ski. Concrètement, il n'y a pas forcément à chaque pas de simulation exactement le nombre de skieurs qui doivent quitter la station, qui atteignent effectivement le bas de la remontée mécanique. C'est pourquoi le nombre d'agents quittant la station à chaque pas est reporté d'un pas à l'autre, et décrémenté chaque fois qu'un agent quitte la station.

Concrètement, à chaque fois qu'un agent arrive au départ d'une remontée mécanique principale, le logiciel teste si la remontée est fermée et en même temps si des agents doivent quitter la station à ce point. Si oui, l'agent quitte la station, si non, l'algorithme teste si des agents sont à supprimer pour ce pas de simulation. Si oui, l'agent quitte la station et le nombre d'agents à supprimer pour ce pas ainsi que le nombre d'agents à supprimer pour cette heure est décrémenté de 1, à cette remontée mécanique. Si non, l'agent rejoint simplement la file d'attente.

Voici le pseudo-code de cette opération : (code exécuté chaque fois qu'un agent arrive au départ d'une remontée mécanique)

```

POUR CHAQUE Remontée mécanique
  SI l'agent se trouve au fond de cette remontée ALORS
    SI la remontée mécanique est fermée
      ET reste des agents à supprimer ALORS
        L'agent quitte la station.
        Le nombre d'agents à supprimer à cette remontée est décrémenté
    FIN SI
    SI reste des agents à supprimer
      ET reste des agents à supprimer pour ce pas ALORS
        L'agent quitte la station.
        Le nombre d'agents à supprimer à cette remontée est décrémenté
        Le nombre d'agents à supprimer durant ce pas est décrémenté
    FIN SI
  SINON
    SI le skieur n'existe pas encore dans la file d'attente
      L'agent rejoint la file d'attente.
    FIN SI
  FIN SINON
FIN SI
FIN POUR CHAQUE

```

## 6.5.7 Plans d'action

### 6.5.7.1 Introduction

Un agent, dans le logiciel Ski Optimal, représente un skieur dans la réalité. Un agent n'est ici pas réellement capable de prendre des décisions de manière autonome. En réalité, cet agent va simplement suivre un plan d'action qui détermine le chemin qu'il va prendre pour se rendre au bas d'une remontée mécanique (voir [Gestion des plans d'action](#)). Petit rappel : un plan d'action est une suite de points décrivant l'itinéraire d'un agent pour se rendre de l'arrivée d'une remontée mécanique au fond d'une remontée mécanique.

### 6.5.7.2 Initialisation des plans d'action

Avant de calculer les plans d'action pour une tranche horaire précise, différentes étapes sont effectuées. Les voici dans l'ordre :

- Le logiciel récupère toutes les données du fichier "VolumeSkiers.dbf", afin de pouvoir les consulter à tout moment (déjà existant dans le logiciel Juste Neige).
- Le logiciel calcule la différence d'agents par rapport à la période précédente.
- Les valeurs sont mises à jour, en fonction des remontées principales (voir [Rectification des comptages SkiData pour une remontée principale](#)).

Ces valeurs sont utilisées afin de définir les probabilités de rejoindre des destinations de piste, lors de la création des plans d'action. La création des plans d'action se fait à l'aide de l'algorithme précédemment développé dans le cadre du logiciel Juste Neige.

Une fois que les différents plans d'action sont créés, le logiciel effectue ces différentes étapes :

- Le logiciel regroupe les plans d'action partant d'un même point de départ.
- Ces différents plans d'action sont mélangés et insérés dans une liste de plans d'action.
- Chaque agent partant du sommet de cette remontée récupère un plan d'action dans la liste des plans d'action et suit cet itinéraire.

Voici ci-dessous le pseudo-code de l'initialisation des plans d'action. Voir aussi [Gestion des plans d'action](#).

```
POUR CHAQUE plans d'action correspondant à une remontée (PAs)
  POUR CHAQUE remontée mécanique
    SI le départ de la remontée = premier point du premier des PAs ALORS
      Rajoute les PAs dans la liste de la remontée.
    FIN SI
    SI la sauvegarde des plans d'action < la liste actuelle ALORS
      Sauvegarde la liste de plans d'action pour
        1. Les dernières remontées
        2. Lorsqu'il n'y a plus de plans d'action disponible
    FIN SI
    SI La remontée mécanique a des plans d'action ALORS
      Mélange la liste des plans d'action
    FIN SI
  FIN POUR CHAQUE
FIN POUR CHAQUE
```

### 6.5.7.3 Gestion des plans d'action

Chaque remontée mécanique se voit attribuer pour chaque heure une liste de plan d'action, et également deux listes de sauvegarde : la sauvegarde des plans d'action utilisée lorsque la remontée mécanique ferme, et la sauvegarde utilisée lorsque la liste de plans d'action est vide. L'utilisation de ces deux listes est décrite ci-dessous.

Nous déterminons que lorsqu'aucun comptage SkiData n'a été relevé pour une heure simulée, la remontée mécanique est fermée. Il arrive donc souvent que des agents soient encore présents dans l'installation-même et dans la file d'attente de celle-ci, lorsque l'installation est considérée comme fermée. Evidemment, il faut que ces agents puissent bénéficier d'un plan d'action une fois arrivés au sommet. C'est à ce moment que la sauvegarde de la liste de plan d'actions, appelée *ListOfActionPlansLastRides*, est utilisée. Cependant, la distribution des plans d'action aux agents se fait avec une petite particularité : seulement les plans d'action avec comme destination des remontées mécaniques ouvertes sont disponibles. Ce fonctionnement suffit, dans la plupart des cas, à faire quitter le domaine à tous les agents. Il subsiste néanmoins une exception, décrite ici (voir [Agents quittant la station](#)).

La vitesse de ski des agents a été déterminée, de manière à approcher le plus possible de la réalité (voir [Evolution des agents](#)). Evidemment, il est très difficile, voir impossible de garantir une simulation reflétant exactement la réalité. Il arrive donc que la liste des plans d'action soit vide avant la fin de l'heure simulée, car les agents ont avancé trop ou pas assez vite. C'est à ce moment que la liste de sauvegarde est utilisée. Lorsque cela se produit, le tracé de l'installation est coloré en rouge (c.f. figure 14), afin d'avertir visuellement le développeur du logiciel que trop d'agents se sont rendus au bas de cette remontée durant l'heure en cours.

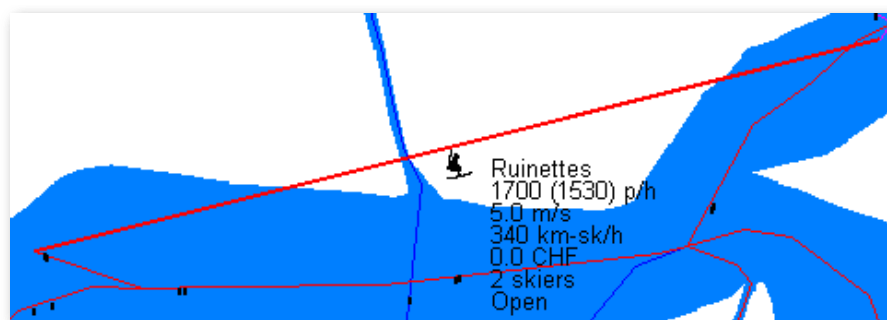


Figure 13 : Liste de plans d'action vide

Dans le futur, lorsque le logiciel intégrera les restaurants sur les pistes de ski et une vitesse de ski davantage réaliste, il y aura la possibilité de réduire ces petites latences, et ainsi de bénéficier d'une simulation encore plus proche de la réalité (voir [Ajout de restaurants et de parkings sur le domaine skiable](#)).

### 6.5.8 Files d'attente

Lorsqu'un agent arrive au bas d'une remontée mécanique, il se place dans la file d'attente de celle-ci, sauf si des agents doivent quitter la station (voir [Agents quittant la station](#)). Chaque départ d'installation a sa file d'attente qui est graphiquement illustrée (c.f. figure 15). On y trouve le nombre d'agents qui attendent et le temps d'attente en minutes. On y trouve également une barre de couleur qui indique graphiquement l'état de la file d'attente, en changeant de couleur en fonction du temps d'attente. Ce dernier est calculé en fonction du débit horaire de la remontée mécanique. Voici les différents états de la file d'attente :

- Temps d'attente inférieur à deux minutes : couleur verte,
- Temps d'attente entre deux et cinq minutes : couleur jaune,
- Temps d'attente entre cinq et dix minutes : couleur orange,
- Temps d'attente supérieur à dix minutes : couleur rouge.



Figure 14 : Différents états files d'attente

### 6.5.9 Encombrement de la station

Durant la simulation, l'utilisateur a la possibilité de mesurer visuellement l'encombrement des pistes de ski. Ceci se fait de deux manières différentes qui sont expliquées ci-dessous.

#### 6.5.9.1 Agents sur le domaine

La simulation multi-agent du logiciel Ski Optimal permet d'afficher les agents se déplaçant sur les pistes de ski. Les agents sont simulés par de petits rectangles noirs. Grâce à cette fonctionnalité, il est possible de se rendre compte visuellement du nombre d'agents empruntant chaque tronçon de piste (voir [Evolution des agents](#)).

#### 6.5.9.2 Couleur des tronçons de pistes

Il est également utile de mesurer visuellement l'encombrement des tronçons de pistes de ski, en fonction de la densité des agents. Cette fonctionnalité prend en compte le nombre d'agents empruntant le tronçon de piste durant l'heure simulée et la largeur de la piste. Ainsi, pour une piste ayant des tronçons où la surface skiable est plus large qu'à d'autres endroits, on aura les tronçons colorés de différentes couleurs, comme l'illustre la capture d'écran ci-dessous.

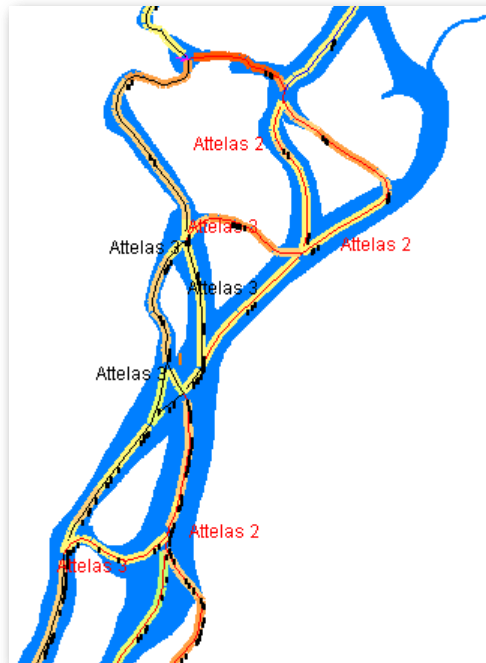


Figure 15 : Densité des agents sur les tronçons de piste

#### 6.5.10 Temps de remontée

Chaque remontée mécanique a une vitesse propre. Cette donnée se trouve dans le fichier "Station.xml", contenu dans le fichier de scénario. En utilisant cette donnée, le logiciel simule le temps que met un agent pour remonter la piste en empruntant cette télécabine. Ce calcul se fait tout d'abord en déterminant à combien de secondes correspond un pas de simulation. Ensuite est calculée la distance parcourue à chaque pas de simulation. Tous les agents présents dans la remontée mécanique avancent donc de  $x$  mètres à chaque pas de simulation. Cette progression est expliquée dans le pseudo-code suivant :

```
POUR CHAQUE remontée mécanique
  SI (la remontée mécanique est ouverte ET
    (il reste des plans d'action OU des plans d'action sauvegardés))
    OU
    (la remontée est fermée mais il reste encore des agents à remonter
    ET il reste des plans d'action pour les dernières remontées) ALORS
    Tous les agents présents dans la remontée avancent de (vitesse de
    La remontée * nb. secondes par pas) mètres
  FIN SI
FIN POUR CHAQUE
```

Lorsqu'un agent entre dans une remontée mécanique, la longueur de la remontée, en mètres, lui est donnée. Lorsque nous parlons d'avancer, il s'agit en fait d'une distance qui est décrétementée pour chaque agent. Il nous reste plus qu'à contrôler, à chaque pas de simulation, combien d'agents ont leur distance plus petite que 1. Ceci nous indique lorsqu'ils sont arrivés au sommet de la remontée mécanique et qu'ils peuvent donc en sortir et récupérer un plan d'action afin de continuer leur progression.



### 6.5.11 Sauvegardes des modifications

Lorsque l'utilisateur rajoute ou modifie une piste, rajoute ou modifie une remontée mécanique, active ou désactive un filtre visuel, ces modifications peuvent être sauvegardées. Pour ce faire, l'utilisateur doit cliquer sur le lien "Save scenario" contenu dans le menu "File". Le raccourci clavier "CTRL+S" peut également être utilisé. Les données sont ensuite sauvegardées au format XML et compressées pour créer le fichier de scénario.

### 6.5.12 Ajout d'une nouvelle remontée mécanique

L'utilisateur a la possibilité de rajouter une nouvelle remontée mécanique dans la station existante. Il dispose pour cela d'une palette d'édition (voir [Edition du théâtre](#)), et d'une fenêtre de personnalisation des données de remontée mécanique (voir [Informations pour une remontée mécanique](#)). La nouvelle remontée doit être convenablement raccordée aux nœuds de la piste pour que la simulation puisse être lancée.

Lors de l'ajout d'une nouvelle remontée mécanique, le logiciel doit pouvoir estimer le nombre d'agents l'empruntant à chaque heure simulée. Pour ce faire, nous devons nous baser sur l'utilisation des autres remontées<sup>32</sup>, afin de calculer des données réalistes. Ce calcul se fait en deux étapes :

- Sauvegarde des rapports d'utilisation des remontées existantes dans la base de données,
- Initialisation des comptages SkiData aux nouvelles remontées mécaniques, en fonction de leur débit respectif et de la moyenne des rapports d'utilisation.

Le pseudo-code suivant illustre ces deux étapes :

```
POUR CHAQUE remontée mécanique
  SI la remontée existe dans le fichier DBF ALORS
    Utilisations += comptages SkiData / Débit réel
  FIN SI
FIN POUR CHAQUE

POUR CHAQUE remontée mécanique
  SI la remontée n'existe pas dans le fichier DBF ALORS
    Nombre de skieurs = débit * moyenne des utilisations
  FIN SI
FIN POUR CHAQUE
```

---

<sup>32</sup> Rapport entre les comptages SkiData et le débit d'une remontée mécanique

### 6.5.13 Création d'un fichier de statistiques

A la fin de la simulation, un fichier de statistiques est généré. Ce fichier est de type Microsoft Excel. Ce choix a été fait, afin de faciliter la réutilisation des données et la création de graphiques par la suite. Le fichier final est nommé de la sorte : « SimulationResults\_YYYYMMJJ\_HHMM.xls ». Il est donc chaque fois différent, étant donné que son nom est composé de l'instant auquel il a été créé.

Lors de la simulation, pour chaque heure, un fichier de statistiques temporaire est créé, sous le nom de "SimulationResults\_temp.xls". Ce fichier est remplacé à la fin de la simulation par le fichier de statistiques complet, comprenant les statistiques de toutes les heures simulées. Le fichier est séparé en deux onglets différents :

Onglet « Statistics » :

- La date et l'heure de la simulation,
- Le type de jour utilisé lors de la simulation,
- Le nom du scénario,
- Les coûts totaux des modifications apportées,
- Pour chaque heure simulée et pour chaque remontée mécanique :
  - Le nom de la remontée mécanique,
  - Le temps d'attente maximal en minutes,
  - Le temps d'attente minimal en minutes,
  - Le temps d'attente moyen en minutes,
  - Le nombre d'agents maximal dans la file d'attente,
  - L'heure à laquelle il y avait le plus d'attente.
- Pour chaque heure simulée, la moyenne des temps d'attente minimal, maximal et moyen.

Onglet « LiftsDatas » :

- La date et l'heure de la simulation,
- Le type de jour utilisé lors de la simulation,
- Le nom du scénario,
- Les coûts totaux des modifications apportées,
- Pour chaque remontée mécanique :
  - Le nom de la remontée mécanique,
  - Le débit nominal en personnes / heure,
  - Le débit réel en personnes / heure,
  - La vitesse de remontée en mètres / seconde,
  - La longueur de la remontée en mètres,
  - Le dénivelé en mètres,
  - Le moment de puissance en kilomètres-skieur / heure

Un exemple de fichier de statistiques créé est annexé à la fin du document présent.

### 6.5.14 Evolution des agents

Les agents, une fois la simulation lancée, évoluent le long des pistes de ski. Comme expliqué plus haut dans le document (voir [Barre de simulation](#)), l'utilisateur a la possibilité de choisir le comportement des agents à l'écran (c.f. figure 16). Les agents suivent soit les pistes manière plus ou moins rectiligne, soit adoptent un comportement proche de la réalité, et oscillent de droite à gauche. L'utilisation de ce comportement n'est malheureusement pas encore conseillée, car le logiciel à ce stade, connaît encore des problèmes de performances. En effet, le calcul de l'oscillation est encore trop demandeur en ressources lors de la simulation. La vitesse de ski des agents oscillant n'est donc pas proche de la réalité.

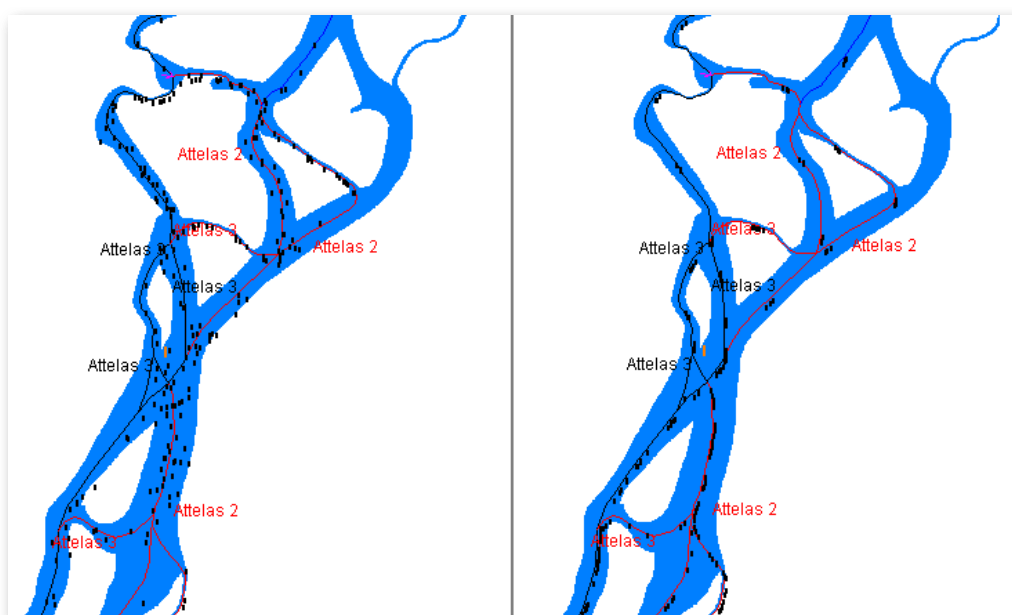


Figure 16 : Agents parcourant les pistes. Avec oscillation (gauche) et sans oscillation (droite)

Selon la vitesse de simulation choisie par l'utilisateur, le nombre d'appels de la méthode de déplacement à chaque pas change. Par défaut, à la vitesse de 100 (100 x la réalité), la méthode "moveSimple", chargée de faire évoluer les agents de manière rectiligne, est appelée 20 fois chaque pas de simulation. La méthode chargée de faire évoluer les agents en oscillant s'appelle "MoveWithOscillation", et est également appelée 20 fois par défaut lorsque l'utilisateur choisit ce mode de comportement.

Ces deux méthodes font partie de la classe SkierPhy.java et ont été développées par M. Marc Revilloud, le développeur du logiciel Juste Neige.

### 6.5.15 Vérification des données SkiData

Lors de l'ouverture d'un scénario, une méthode qui contrôle la validité des comptages SkiData est appelée. Cette fonction vérifie deux choses :

- Si le de comptages SkiData d'une remontée mécanique pour chaque heure est inférieur au débit de celle-ci.
- Le nombre de comptages SkiData au total pour chaque heure est inférieur au nombre de skieurs présents sur le domaine durant cette heure.

Par la suite, d'autres contrôles pourront être rajoutés (voir [Ajout de règles de vérification des données SkiData](#)).

### 6.5.16 Mouvement des remontée mécaniques

Afin de permettre à l'utilisateur de visualiser si une remontée mécanique est en marche, une petite animation a été développée. Lorsque la remontée mécanique fonctionne, chaque deux pas de simulation, le pictogramme de la remontée mécanique bouge de droite à gauche et inversement, comme illustré sur l'image ci-dessous.

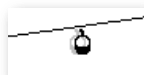


Figure 17 : Visualisation du fonctionnement d'une remontée

## 7 Analyse de performance

Une analyse de performance a été réalisée grâce à l'extension "Eclipse Test and Performance Tools Platform". Cette analyse a permis de démontrer la possibilité d'alléger l'application, notamment en matière de création de nouveaux objets. L'analyse a été effectuée sur la station de Verbier, durant deux heures simulées, à la vitesse de 400x la réalité.

Il en ressort que le fait de créer un nouvel agent, chaque fois que l'un d'entre eux se trouve au bas d'une remontée mécanique, utilise passablement de mémoire. Nous pouvons le constater sur les captures d'écran suivantes.

| Memory Statistics  |                     |                |                     |                 |                   |          |
|--|---------------------|----------------|---------------------|-----------------|-------------------|----------|
| Filter: Default Filter. Click <a href="#">here</a> to set filter |                     |                |                     |                 |                   |          |
| Class Name   | Package             | Live Instances | Active Size (bytes) | Total Instances | <Total Size (b... | Avg. Age |
| int[]  | (default package)   | 14             | 226'218'288         | 62              | 226'223'776       | 0        |
| byte[]   | (default package)   | 5              | 5'243'504           | 4'567           | 15'896'848        | 0        |
| int[][]  | (default package)   | 84             | 2'016               | 86              | 7'105'408         | 0        |
| SkierPhy   | elementOfScenari... | 0              | 0                   | 10'043          | 1'285'504         | 0        |
| Vector2f   | javax.vecmath       | 1'921          | 30'736              | 74'958          | 1'199'328         | 0        |
| Point2i  | javax.vecmath       | 27             | 432                 | 28'336          | 453'376           | 0        |
| Point2f  | javax.vecmath       | 83             | 1'328               | 10'128          | 162'048           | 0        |
| ImageIcon  | javax.swing         | 55             | 2'640               | 3'009           | 144'432           | 0        |
| InsetsUIResource   | javax.swing.plaf    | 38             | 912                 | 4'326           | 103'824           | 0        |
| Edge   | elementOfScenari... | 311            | 42'296              | 494             | 67'184            | 0        |
| ActionPlanPoint  | elementOfScenario   | 1'921          | 46'104              | 1'921           | 46'104            | 0        |
| Type[]   | jxl.biff            | 1              | 496                 | 122             | 31'232            | 0        |
| Label  | jxl.write           | 147            | 8'232               | 512             | 28'672            | 0        |
| CellValue[]  | jxl.write.biff      | 44             | 3'024               | 340             | 15'480            | 0        |
| Number   | jxl.write           | 60             | 3'360               | 228             | 12'768            | 0        |
| Colour[]   | jxl.format          | 1              | 256                 | 62              | 8'432             | 0        |
| StyleXFRecord  | jxl.write.biff      | 20             | 2'720               | 60              | 8'160             | 0        |
| RowRecord  | jxl.write.biff      | 44             | 2'464               | 143             | 8'008             | 0        |
| JMenuItem  | javax.swing         | 13             | 6'136               | 13              | 6'136             | 0        |
| JLabel   | javax.swing         | 15             | 5'880               | 15              | 5'880             | 0        |

Figure 18 : Analyse de performances, avec création de nouveaux skieurs

La capture d'écran ci-dessus montre les performances analysées alors qu'un nouvel agent est créé quand un agent arrive au bas d'une remontée mécanique.

**Memory Statistics**  
Filter: Default Filter. [Click here](#) to set filter

| Class Name       | Package             | Live Instances | Active Size (bytes) | Total Instances | <Total Size (b... | Avg. Age |
|------------------|---------------------|----------------|---------------------|-----------------|-------------------|----------|
| int[]            | (default package)   | 14             | 226'218'288         | 50              | 226'221'072       | 0        |
| int[][]          | (default package)   | 84             | 2'016               | 86              | 7'105'408         | 0        |
| byte[]           | (default package)   | 6              | 5'243'528           | 2'338           | 5'328'288         | 0        |
| Vector2f         | javax.vecmath       | 13'921         | 222'736             | 26'676          | 426'816           | 0        |
| Point2i          | javax.vecmath       | 11'976         | 191'616             | 20'187          | 322'992           | 0        |
| SkierPhy         | elementOfScenari... | 2'000          | 256'000             | 2'000           | 256'000           | 0        |
| Point2f          | javax.vecmath       | 2'083          | 33'328              | 10'073          | 161'168           | 0        |
| ImageIcon        | javax.swing         | 41             | 1'968               | 2'659           | 127'632           | 0        |
| InsetsUIResource | javax.swing.plaf    | 47             | 1'128               | 3'986           | 95'664            | 0        |
| Edge             | elementOfScenari... | 411            | 55'896              | 494             | 67'184            | 0        |
| ActionPlanPoint  | elementOfScenario   | 1'921          | 46'104              | 1'921           | 46'104            | 0        |
| Type[]           | jxl.biff            | 1              | 496                 | 122             | 31'232            | 0        |
| Label            | jxl.write           | 289            | 16'184              | 289             | 16'184            | 0        |
| Number           | jxl.write           | 156            | 8'736               | 156             | 8'736             | 0        |
| Colour[]         | jxl.format          | 1              | 256                 | 62              | 8'432             | 0        |
| CellValue[]      | jxl.write.biff      | 66             | 5'216               | 170             | 8'400             | 0        |
| JMenuItem        | javax.swing         | 13             | 6'136               | 13              | 6'136             | 0        |
| JLabel           | javax.swing         | 15             | 5'880               | 15              | 5'880             | 0        |

Figure 19 : Analyse de performances, sans création de nouveaux skieurs

A l'inverse, la capture d'écran ci-dessus montre les performances lorsque les agents sont réutilisés, et simplement réinitialisés. En résumant ces deux résultats sous forme de tableau (ci-dessous), il ressort nettement que lorsque le logiciel réutilise les objets (agents), il y a logiquement moins d'instances créées, et ainsi les performances générales se voient accrues car moins de mémoire est utilisée.

|          | Objets au total, (nouveaux agents) | Objets au total, (réutilisation des agents) |
|----------|------------------------------------|---|
| Byte[]   | 4'567                              | 2'338                                       |
| SkierPhy | 10'043                             | 2'000                                       |
| Vector2f | 74'958                             | 26'676                                      |

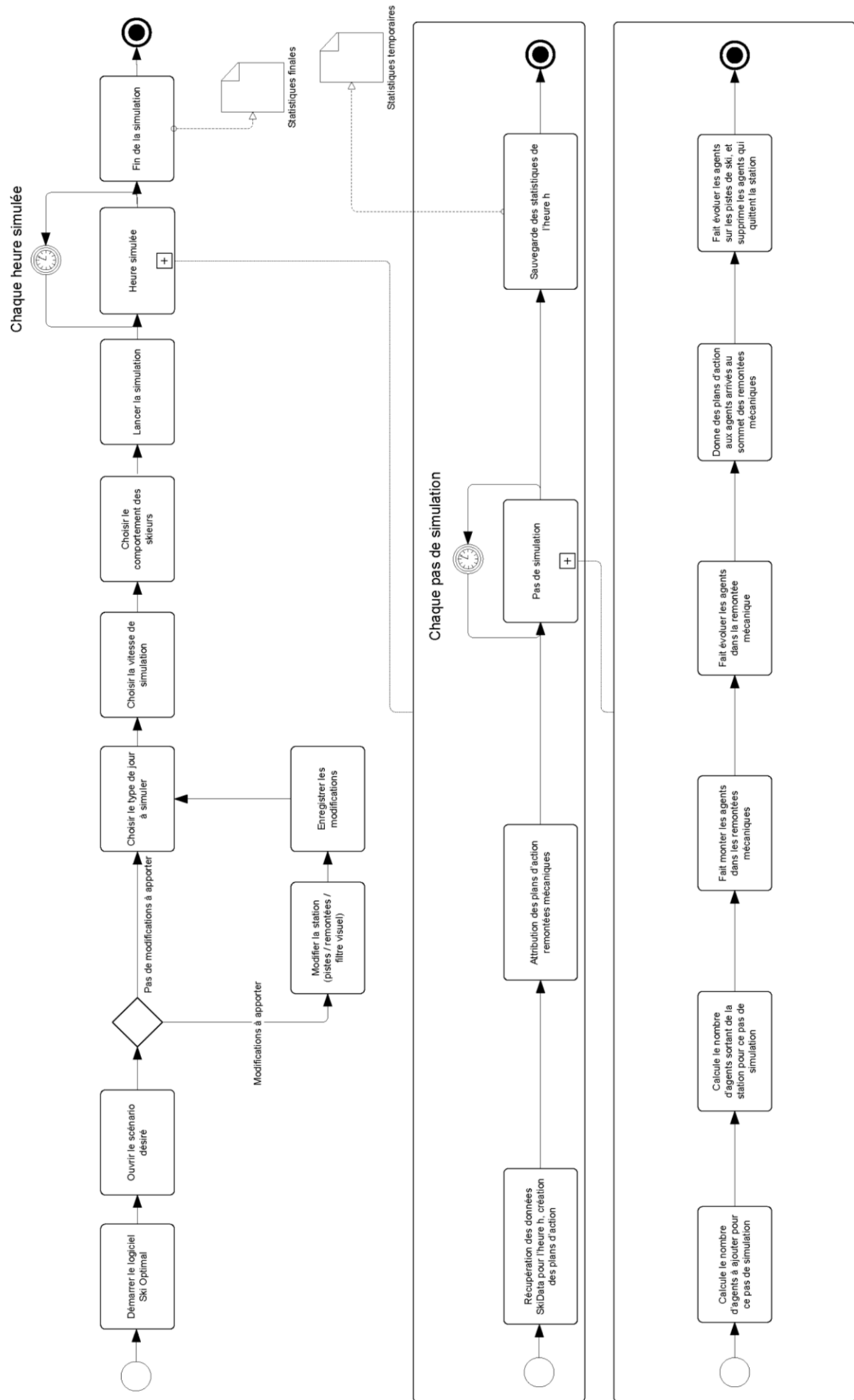
Tableau 3 : Comparatif du nombre d'objets créés, selon le code utilisé

## 8 Utilisation

L'utilisation du logiciel se fait selon les étapes décrites schématiquement à la page suivante. Elle est aisée et simplifiée pour l'utilisateur par des pictogrammes et des raccourcis claviers intuitifs.

Note pour le développeur : lors de la compilation du code dans un environnement du type Eclipse, il convient d'allouer de la mémoire supplémentaire au programme, afin que le chargement du scénario puisse se faire. Pour ce faire, il faut allouer, dans la mesure du possible, au moins 1024 Mo : arguments de la machine virtuelle JAVA : -Xmx1024m.

# Déroulement de la simulation



## 9 Développements futurs

### 9.1.1 Modification du nombre total d'agents lors de l'ajout d'une nouvelle remontée

Lors de l'ajout d'une nouvelle remontée mécanique dans la station, le logiciel n'effectue pour l'instant qu'une estimation du nombre de skieurs empruntant cette remontée. Imaginons que l'utilisateur rajoute plusieurs remontées mécaniques, il arrivera un moment où le nombre d'agents présents au total sur le domaine devra également augmenter. Car nous pouvons facilement imaginer que des nouvelles remontées mécaniques attirent davantage de skieurs sur le domaine skiable. Il faudra donc intégrer cette fonctionnalité.

### 9.1.2 Génération d'un film de simulation

Le logiciel Ski Optimal doit pouvoir offrir la possibilité, en plus des résultats statistiques, de consulter visuellement le déroulement de la simulation de flux.

Au cours de la simulation, des captures d'écran seraient prises à intervalles régulières de (par exemple) 5 minutes simulées, retraçant l'évolution du domaine skiable lors de la journée simulée. Les captures d'écran devront reprendre tout ce que l'utilisateur voit à l'écran. A la fin de la simulation, le logiciel créera un film composé de ces différentes captures d'écran.

### 9.1.3 Insertion des données de comptage horaire par l'utilisateur

Lors de l'ajout d'une nouvelle remontée mécanique sur le domaine (voir [Ajout d'une nouvelle remontée mécanique](#)), l'utilisateur doit avoir la possibilité d'insérer des comptages SkiData "fatices". Actuellement, le logiciel les calcule sur la base des remontées existantes (voir [Modification du nombre total d'agents lors de l'ajout d'une nouvelle remontée](#)).

Etant donné que bon nombre de stations de ski n'utilisent pas de système de comptage de type SkiData, il faut que l'outil permette l'insertion des données par l'utilisateur.

### 9.1.4 Calcul automatique du dénivelé et de la longueur d'une installation

Le logiciel se basant sur des données réelles, il est en mesure de calculer automatiquement la longueur et le dénivelé d'une nouvelle installation mécanique. Ceci est possible grâce à une simple triangulation, et en utilisant notamment fichier d'altitude (voir [Structure d'un scénario](#)). Ce calcul facilitera ainsi à l'utilisateur l'ajout d'une nouvelle remontée mécanique.

### 9.1.5 Ajout de restaurants et de parkings sur le domaine skiable

Dans une station de ski, il arrive un moment où les skieurs se rendent dans les restaurants existant sur le domaine skiable, durant le temps de midi par exemple. Ce comportement modifie le flux et la densité des skieurs sur les pistes de ski. Dans le futur, le logiciel doit pouvoir intégrer de tels établissements, afin de se rapprocher davantage de la réalité.

Nous avons également noté que le fait qu'un départ de remontée mécanique comporte ou non un parking modifie l'attrait de cette remontée mécanique<sup>33</sup>. En effet, cela rend la remontée mécanique "principale", là par où les skieurs rentrent et sortent de la station.

### 9.1.6 Indications sur les pistes

Le logiciel Ski Optimal doit permettre d'ajouter des indications à certains endroits du domaine skiable. Ces indications informent l'agent du temps d'attente pour l'arrivée d'une remontée mécanique, et modifient donc son itinéraire, au fur et à mesure qu'il parcourt les pistes de ski.

---

<sup>33</sup> Lors de la première séance de projet (30.09.10) à Lausanne, M. Jean-Christophe Loubier, qui a déjà développé un logiciel quelque peu similaire, nous a fait remarquer ce point.

### 9.1.7 Coût d'un remblai de piste

Lors du rajout d'une nouvelle remontée mécanique, ou de la modification du débit d'une remontée mécanique existante, il peut arriver que l'on doive élargir une ou plusieurs pistes. Cette opération de remblai a un coût. Ce coût pourrait être calculé en fonction de nombre de mètres/cube de piste modifiés.

Ce calcul pourrait également se faire en exportant le raster au format .dwg, qui serait ensuite interprété par le logiciel de dessin Autocad<sup>34</sup>, afin de calculer ledit coût.

### 9.1.8 Oscillation, performances

A ce jour, le logiciel Ski Optimal intègre la possibilité de faire évoluer les agents de manière rectiligne, ou oscillatoire sur le domaine skiable. L'évolution en forme d'oscillation reste trop gourmande en ressources, et par conséquent ralentit considérablement le déroulement de la simulation.

Il est à noter que l'évolution rectiligne ralentit également le déroulement de la simulation. Certes de manière moindre que le déplacement oscillatoire, mais suffisamment pour que cela ne soit pas agréable pour l'utilisateur. Les deux algorithmes faisant avancer un agent sont donc à revoir, afin de les simplifier et de les rendre plus efficace.

Dans la pratique, il serait éventuellement optimal de ne pas avoir à appeler plusieurs fois la méthode "moveSimple()" sur chaque agent, mais simplement de modifier sa vitesse propre lors du choix de la vitesse de simulation, soit ce champ faisant partie de la classe SkierPhy :

```
private float speed = 1.6f;
```

### 9.1.9 Agents quittant la station

Lorsque des agents quittent la station, l'utilisation des plans d'action est légèrement différente, comparée au reste de la simulation (voir [Gestion des plans d'action](#)). Dans la plupart des cas, les agents quittent complètement la station. Cependant, dans certains cas, l'algorithme actuellement en place ne suffit pas pour faire quitter la station à tous les agents. Comme expliqué plus haut dans le document, lorsqu'une remontée mécanique ferme, seulement les plans d'action avec comme destination une remontée mécanique ouverte sont donnés aux agents. Le problème est que dans certains cas, un agent doit emprunter encore une ou l'autre remontée mécanique avant de quitter la station. Partant du principe que toutes les remontées ferment en même temps, aucun plan d'action ne lui est alors délivré. Ce problème est appelé "cuvette".

Voici un schéma illustrant ce problème, l'explication se trouve en-dessous :

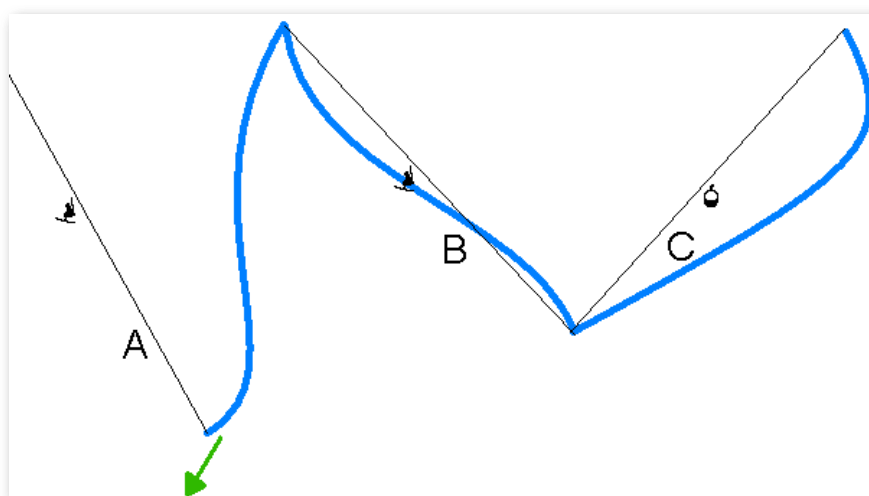


Figure 20 : Illustration du problème de la "cuvette"

<sup>34</sup> Logiciel de dessin par ordinateur, [www.autodesk.fr](http://www.autodesk.fr)



Sur cette illustration, le télésiège A est une remontée mécanique principale. C'est-à-dire que les agents entrent et sortent du domaine skiable par ce point (flèche verte). Lorsque la télécabine C et le télésiège B ferment, les agents se trouvant dans la télécabine C doivent donc encore emprunter le télésiège B pour pouvoir quitter la station. L'algorithme en place ne permet pas ceci, car il délivre aux agents seulement des plans d'action avec comme destination des remontées mécaniques ouvertes.

Un moyen d'éviter ceci serait, comme dans la réalité, de fermer la file d'attente d'une installation une demi-heure avant que celle-ci ne ferme par exemple, et de déterminer les remontées à emprunter pour rejoindre les sorties de pistes de ski. Ainsi, les agents auraient ces remontées comme destination durant la dernière demi-heure avant la fermeture des installations, et quitteraient la station convenablement.

### 9.1.10 Etalonnage du logiciel

Par la suite, il faudra étalonner correctement le logiciel, et notamment la vitesse des skieurs. Le réglage de la vitesse se fait en modifiant la ligne suivante, présente au début du fichier "Simulation.java".

```
public static float skiSpeedBase = 20;
```

L'étalonnage pourra être effectué correctement, une fois que les différents nouveaux modules ont été implémentés, notamment l'ajout de restaurants et de parkings sur le domaine skiable, les performances améliorées, ainsi qu'une meilleure gestion des agents quittant le domaine.

L'étalonnage se fera notamment en vérifiant les temps d'attente obtenus, avec les temps d'attente mesurés dans la réalité. A ce stade, aucune donnée sur les temps d'attente n'est disponible de la part de la station de Verbier (entretien téléphonique avec la station de Verbier, du lundi 8 novembre 2010).

### 9.1.11 Ajout de règles de vérification des données SkiData

Une fonction est appelée à chaque chargement de scénario (voir [Vérification des données SkiData](#)). Cette fonction pourra par la suite être additionnée de nouveaux contrôles de comptages SkiData, en fonction de règles plus précises.

### 9.1.12 Traduction du logiciel

Actuellement, le logiciel n'est traduit qu'en anglais. Cependant, la possibilité de traduire tous les champs est déjà implémentée dans le logiciel et simplement commentée pour l'instant. Cette fonctionnalité est accessible via le menu "Language" de la barre d'outils de l'application.

## 10 Problèmes rencontrés

### 10.1 Déplacement des agents sur les pistes

Lors de la mise en place des nouvelles fonctionnalités, il a fallu modifier le logiciel de base. Ces modifications ont posé des problèmes de performance lors de l'évolution des agents sur les pistes de ski. En effet, la méthode faisant avancer les skieurs est appelée plusieurs fois de suite à chaque pas de simulation, dans le cadre de la simulation Ski Optimal. Cet appel ralentit considérablement les performances de la simulation, c'est pourquoi la méthode en question a été redéveloppée avec la collaboration de M. Marc Revilloud, afin d'être plus efficace et moins gourmande en ressources. Cette méthode n'est néanmoins pas encore complètement efficace, et devra être revue par la suite pour garantir davantage de performances.

## 11 Conclusion

A ce stade du développement, le logiciel Ski Optimal permet à un utilisateur de simuler le flux des agents sur la station de Verbier, durant une journée type. Ce flux s'approche de la réalité car il correspond aux comptages relevés par les bornes SkiData placée au départ de la plupart des installations de la station. Le choix du parcours par un agent se fait suivant des règles de probabilité qui ont été extraites de sondages effectués auprès des skieurs, dans le cadre du développement du logiciel Juste Neige. Il en est ressorti que ce choix dépend le plus souvent de la météo du jour, que peut déterminer l'utilisateur du logiciel Ski Optimal.

L'utilisateur a également la possibilité de modifier la station de ski. Il peut modifier les pistes de ski, leur tracé et leur type, modifier le domaine skiable (zone praticable par les skieurs), modifier ou ajouter une nouvelle remontée mécanique. Il verra ensuite les répercussions de ces changements sur les flux de skieurs et sur les temps d'attente aux départs des différentes installations mécaniques.

Une fois la simulation terminée, l'utilisateur a accès à toutes les données de temps d'attente aux installations. Ces données sont regroupées en un fichier Microsoft Excel. Il peut donc ensuite réutiliser ce fichier afin de le comparer avec d'autres scénarii qu'il aura imaginés.

## 12 Bibliographie et sources

REVILLOUD Marc, 2010 : Rapport final du projet CTI Juste Neige. Rapport final du projet qui a servi de base de développement au projet Ski Optimal.

WIKIPEDIA, 2010 : <http://www.wikipedia.org>. Encyclopédie en ligne, utilisée pour définir les termes suivants : xml, dbf, simulation multi agents.

INTERMAPS, 2010 : <http://ski.intermaps.ch/verbier/index.asp?lang=fr>. Site web présentant le domaine skiable de Verbier, avec notamment les caractéristiques des remontées mécaniques.

LOUBIER Jean-Christophe, 2010 : Use case scénari1. Description des fonctionnalités de l'outil Ski Optimal, ainsi que des différents scénarii possibles.

## 13 Tables des illustrations

|  |    |
|--|----|
| Figure 1 : Sauvegarde d'une version fonctionnelle du code .....  | 7  |
| Figure 2 : Classes avec héritage .....   | 12 |
| Figure 3 : Raster de la station de Verbier .....   | 16 |
| Figure 4 : Représentation du plan des pistes de Verbier .....  | 17 |
| Figure 5 : Palette d'édition du théâtre .....  | 18 |
| Figure 6 : Filtre du théâtre .....   | 19 |
| Figure 7 : Sélection des pistes.....   | 21 |
| Figure 8 : Barre de simulation.....  | 22 |
| Figure 9 : Choix du jour type.....   | 22 |
| Figure 10 : Contrôle de la simulation et choix de la vitesse .....                                     | 22 |
| Figure 11 : Nombre de skieurs, coûts des modifications et choix du comportement des skieurs .....      | 23 |
| Figure 12 : Détails d'une remontée mécanique .....   | 24 |
| Figure 14 : Liste de plans d'action vide.....  | 29 |
| Figure 15 : Différents états files d'attente.....  | 30 |
| Figure 17 : Densité des agents sur les tronçons de piste.....  | 31 |
| Figure 16 : Agents parcourant les pistes. Avec oscillation (gauche) et sans oscillation (droite) ..... | 34 |
| Figure 18 : Visualisation du fonctionnement d'une remontée.....  | 35 |
| Figure 19 : Analyse de performances, avec création de nouveaux skieurs .....                           | 35 |
| Figure 20 : Analyse de performances, sans création de nouveaux skieurs .....                           | 36 |
| Figure 21 : Illustration du problème de la "cuvette" .....   | 39 |

## 14 Table des tableaux

|   |    |
|---|----|
| Tableau 1 : Planning hebdomadaire prévisionnel .....                        | 5  |
| Tableau 2 : Structure du fichier "VolumeSkiers.dbf" .....                   | 15 |
| Tableau 3 : Comparatif du nombre d'objets créés, selon le code utilisé..... | 36 |

## **15 Table des annexes**

À la suite de ce document se trouvent les annexes suivantes :

A. Cahier des charges du projet

B. Statistiques de simulation

# Annexe

## A. Cahier des charges du projet

# Annexe

## B. Statistiques de simulation